TECHNICAL REPORT RD-82-19

C.1

AN APPROACH TO EXPERIMENTAL INVESTIGATION OF JET PLUME EFFECTS ON MISSILE AERODYNAMICS

George M. Landingham
Systems Simulation and
Development Directorate
US Army Missile Laboratory

July 1982

U.S. ARMY MISSILE COMMAND
Redstone Arsenal, Alabama 35809

Approved for public release; distribution unlimited.

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: RD-82-19	2. GOVT ACCESSION NO.
	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): An Approach to Experimental Investigation of Jet Plume Effects on Missile Aerodynamics	5. TYPE OF REPORT & PERIOD COVERED: Technical Report
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): George M. Landingham	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Commander, US Army Missile Command, ATTN: DRSMI-RD, Redstone Arsenal, AL 35898	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Commander, US Army Missile Command, ATTN: DRSMI-RPT, Redstone Arsenal, AL 35898	12. REPORT DATE: July 1982
	13. NUMBER OF PAGES: 85
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report): Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Plume Modeling
Plume Effects
Wind Tunnel Testing

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

An approach to modeling a rocket's plume effects based on the theory of Korst is presented. To implement the modeling scheme an interactive Fortran program was developed which designs model nozzles that produce geometrically similar plumes with similar base flow characteristics as prototype nozzles but use air or some other medium instead of a propellant. Included are the modeling theory, experimental results and the Fortran program with a sample case.

TABLE OF CONTENTS

		Page No.
I.	INTRODUCTION	5
II.	THEORY OF JOHANNESEN AND MEYER	6
III.	KORST'S PLUME MODELING THEORY	14
IV.	NOZZLE SOLUTION BY METHOD OF DUTTON AND ADDY WITH METHOD OF CHARACTERISTICS	16
V.	PLUME MODELING EXPERIMENTS	19
	APPENDIX I. SAMPLE CASE	24
	APPENDIX II. MODEL NOZZLE DESIGN PROGRAM LISTING	28
	REFERENCES	84

SYMBOLS

Sections II and III

a	local speed of sound
a^*	critical speed of sound
M	Mach Number
M^*	critical Mach number
q	velocity magnitude
r_c	initial radius of curvature of expanded free jet boundary
R	ratio of distance from center of expansion to distance of center of axis of symmetry
u, v	velocity components
y	Cartesian coordinates
$\beta_o, C_1, A_1, K_1, \beta_2$	constants of integration
γ	specific heat ratio
ϕ	polar coordinate
λ	$((\gamma-1)/(\gamma+1))^{\frac{1}{2}}$
μ	local Mach angle
η	auxiliary angle
Δ	angle at the center containing the region of centered expansion
δ	angle of deflection of the streamline at the center of expansion
θ	streamline angle
ψ	$\phi-\beta_1$

SYMBOLS

Section IV

a^*	sonic velocity
M	Mach number
M^*	critical Mach number
R	normalized throat radius of curvature, $R=Rw/r^*$
Rw	nozzle throat radius of curvature
r^*	nozzle throat radius
u,v	velocity components
X,Y	Cartesian coordinates
z,r	normalized coordinate of Hall's expansion
η	Dutton-Addy expansion variable
θ	streamline angle

Subscripts

c	point in nozzle where the circular throat joins the conical section
F	conditions at final expansion fan line as $R \to 0$
M	model

Subscripts

F	conditions at final expansion fan line as $R \rightarrow 0$
L	conditions at initial expansion fan line as $R \rightarrow 0$
M	model
N	limiting value as the center of the expansion is approached along the nozzle boundary streamline
P	prototype

I. INTRODUCTION

The interaction of a rocket's plume with the external flow can significantly affect the aerodynamic performance of the rocket. The adverse effects are well known and include increased drag, base heating and plume induced separation. There is at least one instance where plume interaction can be beneficial. Some rocket configurations induce large stable pitching moments in the transonic flight regime which cause the vehicle to be overly wind sensitive. The interaction between the slipstream and an underexpanded plume can act to de-stabilize the rocket in this region.

Whether to avoid the adverse conditions or to take advantage of beneficial effects, the missile designer needs methods of investigating the plume-slipstream interaction. In the transonic region, experimental testing is the primary method of obtaining data and it is most desirable to have wind tunnel models which simulate the proper plume characteristics without actual propellant burning. Korst[1] has presented a method for designing models of conical nozzles which produce the same geometrical plume as the prototype nozzles while accounting for significant viscid and inviscid aspects of the base flow problem. The flow analysis used by Korst is based on concepts developed by Johannesen and Meyer[2] which allow the flow field near the centered expansion at the nozzle's exit to be expressed in the form of a series expansion with respect to the radius vector. For a set of specified prototype nozzle conditions, Korst's modeling theory can be used to calculate model nozzles with the proper combination of exit lip Mach number, wall flow acceleration and lip wall angle. For nozzles with nominal divergent sections where source flow is approximated the modeling requirements are reduced to determining the exit Mach number and exit wall angle. This modeling concept has also been presented in a paper by Korst and Deep[3] which suggests a numerical solution for one of the approaching flow velocity terms as an alternative to the closed form solution of Johannesen and Meyer.

This latter scheme, while determining the requisite exit conditions, does not account for the details of the nozzle geometry necessary to produce these exit conditions. For a conical nozzle with a circular arc throat section the geometrical quantities which must be determined are throat radius of curvature, nozzle length and the axial position where the circular arc section and conical divergent section join. To evaluate the effects of throat radius of curvature use is made of the transonic throat flow approximations originally proposed by Hall[4] and corrected by Kliegel and Levine[5] and later refined by Dutton and Addy[6]. The flow field as calculated by this method then provides starting conditions for a method of characteristics routine which solves the flow field in a radial plane fashion until the exit conditions specified by Korst's modeling theory are matched. The point where the exit conditions match is then the nozzle length. The point where the circular arc section and conical section join can be determined from purely geometrical considerations.

Three programs written by Prof. Korst in BASIC for the Hewlett-Packard 9830 were re-written in FORTRAN by the author and combined into a single interactive model design program. This report presents an overview of the Johannsen and Meyer theory, the throat flow approximation theory and Korst's modeling theory. The report also illustrates how the three theories are used to develop the model design program. The program also calculates plume shape

using a method of characteristics routine which is initialized by the exit conditions determined from the modeling. The FORTRAN program is included with user instructions and a sample case.

II. THEORY OF JOHANNESEN AND MEYER

The Korst plume modeling theory is based on the theory of Johannesen and Meyer[2] which provides an approximate method for solving the flow field at the lip of an axially-symmetric nozzle. The approach assumes the flow is a perfect gas that is isentropic, irrotational and steady.

When a uniform, two-dimensional flow with these properties expands around a corner it is called Prandtl-Meyer flow and is characterized by generating centered simple waves. In simple-wave flow the disturbances created by the corner are propagated as pressure waves along the Mach lines. Further, the Mach lines are assumed straight and the flow properties along the Mach lines are uniform. The velocity magnitude at any point in a centered wave is characterized by streamlines such that all the Mach lines pass through a common origin. When the flow is axially-symmetric, the expansion is no longer simple, though it is still centered.

The method proposed by Johannesen and Meyer solves the flow field in the neighborhood of the lip by assuming that the flow may be divided into three regions, a centered expansion occurring at the lip, and flow fields before and after the expansion. A polar coordinate system with variables R and ϕ is employed, see Figure 1, where R is the ratio of the distance from the center of the expansion to the distance of the center of the axis of symmetry, such that the exit radius of the nozzle has a value of 1. The polar coordinate, ϕ, is measured from the exit plane of the nozzle. A series expansion in powers of R gives the following velocity components:

$$u(R,\phi) = u_o\ (\phi) + Ru_1\ (\phi) + R^2u_2\ (\phi) + \ldots \tag{1}$$

$$v(R,\phi) = v_o\ (\phi) + Rv_1\ (\phi) + R^2v_2\ (\phi) + \ldots \tag{2}$$

The first term in this expansion gives a solution identical to the two-dimensional solution, that is a uniform flow field before and after the expansion and a Prandtl-Meyer flow field in the region of the expansion. The equations for the second term in the series must be integrated to provide a solution. With the resulting equations and the proper application of boundary conditions, the initial curvature of a jet boundary can be calculated with an error of $O(R^2)$.

With the further assumption that the expansion at the lip is shock-free, the development of Johannesen and Meyer is presented in the following paragraphs.

The polar equations of continuity and of motion may be combined to give

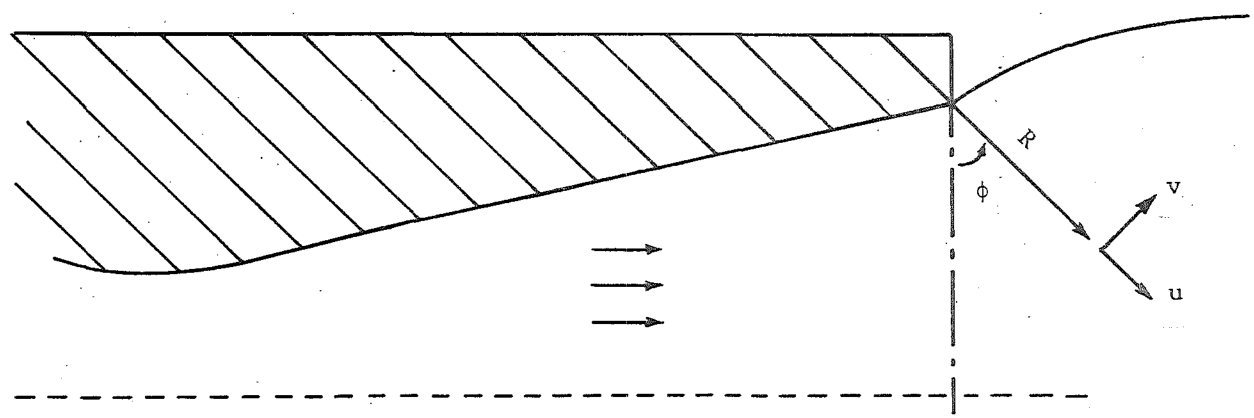

Figure 1. Axisymmetric coordinates used in Johannesen and Meyer expansion.

$$(u^2 - a^2)\frac{\partial u}{\partial R} + (v^2 - a^2)\frac{\partial v}{R\partial\phi} + uv\left(\frac{\partial v}{\partial R} + \frac{\partial u}{R\partial\phi}\right) - a^2\frac{u}{R}$$

$$- \frac{a^2}{1-R\cos\phi}(v\sin\phi - u\cos\phi) = 0 \quad . \tag{3}$$

Bernouilli's equation is

$$u^2 + v^2 + \frac{2}{\gamma-1}a^2 = \frac{a^{2*}}{\lambda^2} \tag{4}$$

where a is local speed of sound, a* is critical speed of sound, γ is specific heat ratio and $\lambda = \sqrt{\frac{\gamma-1}{\gamma+1}}$. The irrotationality of the flow is expressed by:

$$\frac{\partial u}{R\partial\phi} - \frac{1}{R}\frac{\partial(Rv)}{\partial R} = 0 \quad . \tag{5}$$

If the series representation of the velocity components as given by (1) and (2) are substituted into equations (4) and (5), coefficients of various powers of R are equated and $O(R^2)$ and greater are discarded the following equations are obtained:

$$u_o^2 + v_o^2 + \frac{2}{\gamma-1}a_o^2 = \frac{a^{2*}}{\lambda^2} \tag{6}$$

$$u_o u_1 + v_o v_1 + \frac{2}{\gamma-1}a_o a_1 = 0 \tag{7}$$

$$v_o = u_o' \tag{8}$$

$$v_1 = \frac{1}{2}u_1' \tag{9}$$

where $u_o' = \frac{du_o}{d\phi}$ and $u_1' = \frac{du_1}{d\phi}$.

If equation (3) is expanded with the series representations of u and v, $O(R^2)$ and greater are again discarded, powers of R are collected and equations (6) to (9) are used the following two equations can be derived:

$$(u_o + v_o')(v_o^2 - a_o^2) = 0 \tag{10}$$

$$(v_o^2 - a_o^2)(v_1' + u_1) + (u_o + v_o')[(\gamma-1)u_o u_1 + (\gamma+1)v_o v_1] + 2u_o v_o v_1$$

$$+ (u_o^2 - a_o^2)u_1 - a_o^2(v_o\sin\phi - u_o\cos\phi) = 0 \quad . \tag{11}$$

From equation (10) it is seen that there are two solutions to the system of equations. One solution approaches uniform flow and the other solution approaches a Prandtl-Meyer expansion as $R\to 0$.

A. The Centered Expansion.

To arrive at equations for the centered expansion let

$$v_o^2 - a_o^2 = 0 \tag{12}$$

then from (6) and (8) after integrating

$$u_o(\phi) = \frac{a^*}{\lambda} \sin \left[\lambda\ (\phi + \beta_o)\right] \tag{13}$$

$$v_o(\phi) = a^* \cos \left[\lambda\ (\phi + \beta_o)\right] \tag{14}$$

where the constant, β_o, will be determined by satisfying the initial conditions of the approaching flow.

Equation (11) can be reduced by substituting in equation (12) after rearranging equation (11) becomes:

$$2u_o u_1' + \left[\frac{3\gamma-1}{\gamma+1}\left(\frac{u_o}{v_o}\right)^2 - 1\right] v_o u_1 - v_o(v_o \sin\phi - u_o \cos\phi) = 0 \quad . \tag{15}$$

Then after substituting (13) and (14) into (15) and integrating, Johannesen and Meyer derived the following:

$$u_1(\phi) = \frac{-a^*}{2\lambda} (\cos\eta)^{\frac{1}{2}\left(\frac{3\gamma-1}{\gamma-1}\right)} \mathrm{X} \ (\sin\eta)^{\frac{1}{2}}$$

$$\mathrm{x}\left\{I_1(\eta)\cos\beta_o + I_2(\eta)\sin\beta_o - \lambda I_3(\eta)\cos\beta_o + \lambda I_4(\eta)\sin\beta_o + C_1\right\} \tag{16}$$

where $\eta = \lambda\ (\phi + \beta_o)$ and C_1 is an arbitrary constant of integration, and

$$I_1(\eta) = \int_{\eta i}^{\eta} \cos\ (y/\lambda)\ (\sin y)^{-\frac{1}{2}}\ (\cos y)^{-\frac{1}{2}\lambda^2}\ dy$$

$$I_2(\eta) = \int_{\eta_i}^{\eta} \sin\ (y/\lambda)\ (\sin y)^{-\frac{1}{2}}\ (\cos y)^{-\frac{1}{2}\lambda^2}\ dy$$

$$I_3(\eta) = \int_{\eta_i}^{\eta} \sin\ (y/\lambda)\ (\sin y)^{-3/2}\ (\cos y)^{\frac{1}{2}\frac{\gamma-3}{\gamma-1}}\ dy$$

$$I_4(\eta) = \int_{\eta_i}^{\eta} \cos\ (y/\lambda)\ (\sin y)^{-3/2}\ (\cos y)^{\frac{1}{2}\frac{\gamma-3}{\gamma-1}}\ dy$$

where y is the Cartesian coordinate and the subscript i indicates the beginning of the expansion region. Then from equations (9) and (15):

$$v_1(\phi) = \frac{\frac{1}{2}\, a^*\, \frac{v_o}{a^*}\left[\frac{v_o}{a^*} \sin\phi - \frac{u_o}{a^*} \cos\phi\right] + \frac{v_o}{a^*}\ \frac{u_1}{a^*}\left[1 - \frac{3\gamma-1}{\gamma+1}\ \frac{u_o}{v_o}\right]^2}{2\ \frac{u_o}{a^*}} \tag{17}$$

This approach by Johannesen and Meyer can be simplified by the use of a numerical integration technique such as Runge-Kutta to solve u_1 (ϕ) directly from equation (15).

B. The Approaching Flow Solution.

Beginning with

$$u_o + v_o' = 0 \tag{18}$$

and following a similar procedure the following equations were derived:

$$u_o (\phi) = w\cos\Psi \tag{19}$$

$$v_o (\phi) = w\sin\Psi \tag{20}$$

$$u_1 (\phi) = A_1 \cos (2\Psi + \beta_2) + K_1 \tag{21}$$

$$v_1(\phi) = -A_1 \sin (2\Psi + \beta_2) \tag{22}$$

where $\phi - \Psi$ = constant = β_1 and A_1, K_1 and β_2 are constants.

To solve for the initial curvature of a jet boundary, it is necessary to consider the geometry of the flow at the lip, see Figure 2. In Figure 2, ϕ_N is the angle between the nozzle wall and the exit, ϕ_L and ϕ_F are the respective angles of the initial and final Mach lines of the expansion region as $R \to 0$. Then

$$\phi_L = \pi - \mu_L + \phi_N \tag{23}$$

$$\phi_F = \Delta + \phi_L \tag{24}$$

$$\phi_j = \quad _F + \phi_F \tag{25}$$

$$\Delta = \delta + \mu_L - \mu_F \tag{26}$$

where μ_L and μ_F are the Mach angles bounding the expansion, Δ is the angle containing the expansion and δ is the angle of deflection of the boundary streamline.

The boundary conditions along the wall of the nozzle are:

$$u_o (\phi_N) = -q_L \tag{27}$$

$$v_o (\phi_N) = 0 \tag{28}$$

where q is the velocity mangitude, $q = \sqrt{u_o^2 + v_o^2}$ in the uniform flow region. Then by equation (20)

$$\Psi = \phi - \phi_N \tag{29}$$

Then from equations (23) through (26)

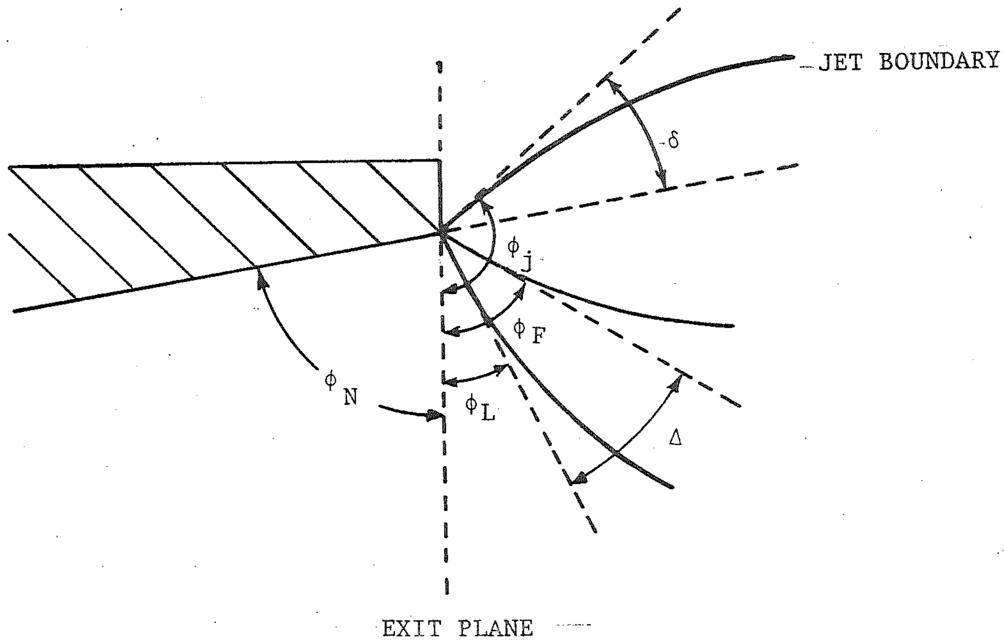

Figure 2. Flow geometry at nozzle exit.

$$u_o(\phi_L) = q_L \cos \mu_L \tag{30}$$

$$v_o(\phi_L) = q_L \sin \mu_L \quad . \tag{31}$$

Combining with equations (21) and (22) gives

$$u_1(\phi_L) = 2 \cos^2\mu_L u_1(\phi_N) - \sin 2\mu_L u_1(\phi_N) + \frac{a_o^2}{q_L} \cos (\phi_N) \quad . \tag{32}$$

From Figure 2 the angle of the streamline is given by

$$\theta = \phi - \frac{\pi}{2} + \tan^{-1}\left(\frac{v}{u}\right) \quad . \tag{33}$$

In the uniform flow region $\frac{\partial\theta}{\partial\phi} = 0$ and the boundary conditions at the wall at R=0 is $v(o,\phi_N) = v_o(\phi_N) = 0$, then the curvature of the streamline at the origin is given by:

$$\left.\frac{d\theta}{dR}\right|_{R=0} = \frac{v_1(\phi_N)}{u_o(\phi_N)} \tag{34}$$

also for the velocity, q

$$q \frac{dq}{dR} = q \left(\frac{\partial q}{\partial\phi}\right)\left(\frac{d\phi}{dR}\right) + q\frac{\partial q}{\partial R} \tag{35}$$

but $\frac{\partial q}{\partial\phi} = 0$,

$$q \frac{dq}{dR} = u_o(\phi_N)u_1(\phi_N) \tag{36}$$

and equation (32) becomes

$$u_1(\phi_L) = q_L \sin 2\mu_L \left[\left.\frac{d\theta}{dR}\right|_{R=0} - \frac{1}{q_L} \cot \mu_L \left.\frac{dq}{dR}\right|_{R=0}\right] - q_L \sin \theta_N \sin^2\mu_L \quad . \tag{37}$$

The geometry also gives

$$\lambda\Delta = \cot^{-1}\left(\frac{\tan \mu_F}{\lambda}\right) - \cot^{-1}\left(\frac{\tan \mu_L}{\lambda}\right) \quad . \tag{38}$$

The constant β_o is determined in the following manner: From equation (13) and (8)

$$v_o = u_o' = \frac{a^*}{\lambda} \cos [\lambda(\phi + \beta_o)]\lambda$$

then dividing equation (31) by (30)

$$\tan \mu_L = \frac{v_o\ (\phi_L)}{u_o\ (\phi_L)}$$

then equating the two sets of relationships for the velocity components

$$\tan\ \mu_L = \frac{v_o(\phi_L)}{u_o(\phi_L)} \Bigg| = \tan\ \mu_L$$

from which

$$\beta_o = -\phi_L + \frac{1}{\lambda} \tan^{-1} \left[\frac{\lambda}{\tan\ \mu_L}\right] \tag{39}$$

and equation (4) gives a relation for q:

$$\left(\frac{q_F}{q_L}\right)^2 = \frac{\gamma-1 + 2\ \sin^2\ \mu_L}{\gamma-1 + 2\ \sin^2\ \mu_F} \ . \tag{40}$$

The downstream uniform flow field imposes the boundary conditions that $u_o(\phi_j) = q_F$, $v_o\ (\phi_j) = 0$ so that $\beta_1 = \phi_j$ and the final velocity in the expansion region is q_F. Again using the uniform flow equations the following relationship for the initial curvature of the jet boundary is derived:

$$\left.\frac{d\theta}{dR}\right|_{R=0} = \frac{1}{q_F} \cot\ \mu_F \left.\frac{dq}{dR}\right|_{R=0} - \frac{q_F \sin\ \theta_j\ \sin^2\ \mu_F + u_1(\phi_F)}{q_F\ \sin\ \mu_F} \ . \tag{41}$$

The Johannesen and Meyer theory can be used in plume modeling since it allows the determination of the initial radius of curvature of the plume boundary. The next section will illustrate how Korst takes advantage of this theory to match plumes from two geometrically different nozzles. To summarize the procedure for obtaining a value of initial curvature of the jet boundary the following steps are listed:

(1) The flow conditions approaching the lip are known, that is Θ_N, $\left.\frac{dq}{dR}\right|_{R=0}$ and $\left.\frac{d\Theta}{dR}\right|_{R=0}$.

(2) μ_L and $u_1\ (\phi_L)$ are calculated from the known flow conditions and equation (37).

(3) q_F is calculated based on known outside pressure.

(4) μ_F and Δ are calculated from (40)and (38) through (39) respectively.

(5) δ is calculated from (30).

(6) The integrals of equation (16) are evaluated between the limits

$\eta_L = \cot^{-1} \dfrac{\tan\mu_L}{\lambda}$ and $\eta = \eta_L + \lambda\Delta$.

(7) Equation (16) is evaluated for u_1 (ϕ_F). Or equation (15) is integrated numerically.

(8) Finally equation (41) can be evaluated to obtain the initial curvature of the jet boundary.

III. KORST'S PLUME MODELING THEORY

The Johannesen and Meyer theory described in the previous section provides a method for determining the initial radius of curvature and initial slope of the jet boundary of a rocket nozzle. These two parameters are sufficient to describe the initial geometry of the plume. Korst[1] expands on this idea by asserting that to properly model a prototype nozzle the plume geometry of the prototype and model must be the same. Korst has developed a numerical interation technique which uses the theory of Johannesen and Meyer to first determine the radius of curvature and slope for a specified prototype and then to determine the exit flow conditions of a model which geometrically matches the prototype plume. It is generally assumed that the specific heat ratio of the model and prototype are not the same. This means that in wind tunnel testing, air or some other inert gas, such as Freon, can be used rather than the actual rocket propellant.

Consider the velocity equations of Section II as non-dimensionlized by the critical speed of sound, a*. For the flow approaching the nozzle exit, equations (19) and (20) become respectively:

$$\frac{u_o(\phi_L)}{a^*} = M_L^* \cos \mu_L \tag{42}$$

$$\frac{v_o(\phi_L)}{a^*} = M_L^* \sin \mu_L \tag{43}$$

where M* is the critical Mach number. Equation (37) which gives the relation for the expansion term in a uniform flow region may be written:

$$\frac{u_1(\phi_L)}{a^*} = M_L^* \{\sin 2\mu_L \left[\frac{d\theta}{dR}\Big|_{R=0} - \frac{1}{M_L^*}\frac{dM^*}{dR}\Big|_{R=0} \cot \mu_L\right] - \sin\theta_L \sin^2 \mu_L\}. \tag{44}$$

For a conical nozzle, Θ = constant, $\frac{d\Theta}{dR} = 0$ so that equation (44) reduces to

$$\frac{u_1(\phi_L)}{a^*} = M_L^* \frac{4 \sin \theta_L \cos^2\mu_L}{\frac{2}{\gamma-1} \frac{\lambda^2 M_L^{*2}}{1-\lambda^2 M_L^*} - 1} \quad . \tag{45}$$

The uniform flow approaching the nozzle lip is also the initial flow for the expansion region. The constant C_1 in equation (16) can be evaluated since $\eta = \eta_L$ so that the integrals vanish, μ_L can be determined from the approaching flow, ϕ_L can be evaluated from equation (23) so that β_o is obtained from equation (39).

Equation (41) can also be re-written in terms of M* as:

$$\left.\frac{d\theta}{dR}\right|_{R=0} - \frac{1}{M_F^*} \left.\frac{dM^*}{dR}\right|_{R=0} \cot\mu_F = \frac{-1}{\sin2\mu_F} \left[\sin\theta_F \sin^2\mu_F + \frac{u_1(\phi_F)}{a^*} \frac{1}{M_F^*}\right] \tag{46}$$

from which the desired result, the initial radius of curvature of the jet boundary, r_c, can be obtained, since:

$$r_c = \frac{1}{\left.\frac{d\theta}{dR}\right|_{R=0}} \quad . \tag{47}$$

The assumption of a conical nozzle also defines the approaching streamline angle at the nozzle wall as being equal to the conical divergence angle, Θ_L. Therefore, the Prandtl-Meyer relations can be used to find the final streamline angle, Θ_F, since:

$$\theta_F = \theta_L + \omega(M_F) - \omega(M_L) \tag{48}$$

and

$$\phi_F = \theta_F - \mu_F + \frac{\pi}{2} \quad . \tag{49}$$

Geometric modeling of the plume is achieved when the initial slope of the jet boundary of the model and prototype and the plume radius of curvature of the model and prototype are the same. That is when:

$$\Theta_{F_M} = \Theta_{F_P} \tag{50}$$

and

$$r_{C_M} = r_{C_P} \quad . \tag{51}$$

It is assumed that for the prototype all the specifying parameters are known. For the model, the choice of a propellant gas fixes γ_M. The numerical procedure used by Korst is essentially as outlined at the end of Section II as simplified by the conical nozzle assumption, $\frac{d\Theta}{dR} = 0$. By selecting a value of M_{L_M} there is obtained a value for Θ_{L_M} since by equation (50) Θ_{F_M} is known and equation (48) can be used provided M_{F_M} is known. By following the calculation procedure a value for r_{C_M} is obtained. r_{C_M} is compared with a value of r_{c_p} calculated from the prototype conditions, the value of M_{L_M} is adjusted accordingly and the calculation procedure continues until $r_{C_M} = r_{c_p}$.

Before the iteration process can begin, M_{F_M}, which is dependent on the external flow, must be determined by some method. Korst refers to the calculation of this parameter as the closure condition for the wake. There are several possibilities for the choice of a closure condition relating to the recompression ratio at the end of the wake or to the conservation of mass in the wake; however, with only one unresolved parameter, both conditions cannot be satisfied simultaneously. Korst chooses to match the inviscid streamline deflection pressure rise as given by

$$\frac{\gamma_P \; M^2{}_{F_P}}{\sqrt{M^2_{F_P} - 1}} = \frac{\gamma_M \; M^2{}_{F_M}}{\sqrt{M^2_{F_M} - 1}} \tag{52}$$

for a weak shock recompression or

$$\frac{2\gamma M^2_{FM} - (\gamma_M - 1)}{\gamma_M + 1} = \frac{2\gamma_P M^2_{FP} - (\gamma_P - 1)}{\gamma_P + 1} \tag{53}$$

for a strong shock recompression.

Since M_F is dependent on the jet to ambient pressure ratio, the model nozzle is designed for a specific Mach number and altitude, the "design point". This procedure calculates a value for the model nozzle conical divergence angle, Θ_{F_M}, but does not determine throat radius of curvature and nozzle length. The method utilized to calculate these parameters is discussed in Section IV.

IV. NOZZLE SOULTION BY METHOD OF DUTTON AND ADDY WITH METHOD OF CHARACTERISTICS

Korst's plume modeling theory as described in Section III provides a method of determining the exit Mach number and conical divergence angle for a model nozzle which has the same initial jet boundary radius of curvature and slope as a known prototype nozzle. This section discusses a procedure for determining a length and throat radius of curvature for the model nozzle which produces the exit conditions. The procedure utilizes an expansion method

developed by Dutton and Addy[6] to solve the transonic flow in the region of the nozzel throat which establishes initial conditions for a method of characteristics program. The method of characteristics routine solves the flow in the expanding nozzle until the Mach number at the wall boundary is equal to the final Mach number, M_{F_M}, determined by the modeling program.

Because the equations for the method of characteristics are dependent upon the ralationship $\sqrt{M^2-1}$, other methods must be used to solve the transonic flow region in convergent-divergent nozzles. Hall[4] developed a widely used expansion technique which is a small-perturbation method from the one-dimensional flow solution. Hall's solution in cylindrical coordinates is given by

$$u = 1 + u_1\,(r,\ z)\,\varepsilon + u_2\,(r,\ z)\,\varepsilon^2 + u_3\,(r,\ z)\,\varepsilon^3 + \ldots \tag{54}$$

$$v = \left(\frac{\gamma}{2}\,\varepsilon\right)^{1/2} \left[v_1\,(r,\ z)\,\varepsilon + v_2\,(r,\ z)\,\varepsilon^2 + v_3\,(r,\ z)\,\varepsilon^3 + \ldots\right] \tag{55}$$

where u and v are axial and radial velocity components normalized by the sonic velocity, a*. The expansion variable ε is defined to be 1/R. R is the normalized throat radius of curvature, R=Rw/r*, where Rw is the actual throat radius of curvature and r* is the nozzle throat radius. The transformed normalized axial coordinate, z, and the normalized radial coordinate, r, are defined by $z = \left(\frac{2R}{\gamma+1}\right)^{1/2} \frac{x}{r^*}$ and $r = \frac{y}{r^*}$.

Hall's equations, (54) and (55) are well-behaved provided R>1.5. Kliegel and Levine[5] had seemingly overcome this limitation by using a series expansion where $\varepsilon = 1/(R+1)$ for axisymmetric nozzles. Kliegel and Levine developed their series solution in torodial coordinates such that the axis is represented by a line $\eta = 0$ and the nozzle wall is represented by a line $\eta = \eta_w$. The results were then transformed back to cylindrical coordinates. Unfortunately, the series do not satisfy the governing differential equations of motion in cylindrical coordinates. This paper by Kliegel and Levine[5] is still important in that the authors re-derived Hall's original equations and corrected errors in the third order terms.

Dutton and Addy[6] formulated the problem in a similar manner to Hall except that the expansion veriable, $\varepsilon = 1/(R+\eta)$ was used. Details of the series derivation can be found in references 4 and 6 and will not be presented here. The basic approach, however, is to substitute equations (54) and (55) into the governing equations and into the boundary conditions. The governing equations are taken to be the gas dynamic equation and the irrotationality condition. The boundary conditions are that both the axis of symmetry and the nozzle wall are streamlines. After substituting, coefficients of like powers of E are gathered to formulate the various orders of the expansion. The equations formulated in this manner are then solved by assuming solution forms suggested by the boundary conditions.

Dutton and Addy suggests using third order, $\eta = 1$ solutions for most applications. This is the approach taken in the program presented in this report. The equations for the third order velocity terms, the non-dimensional Mach number and the streamline deflection angles are given as follows:

$$u_1\ (r,z) = \tfrac{1}{2}\, r^2 - \tfrac{1}{4} + z \tag{56}$$

$$u_2\ (r,z) = \frac{2\gamma+9}{24} r^4 - \frac{4\gamma+15-12\eta}{24} r^2 + \frac{10\gamma+57-72\eta}{288} + \left(r^2 + \frac{4\eta-5}{8}\right) z - \left(\frac{2\gamma-3}{6}\right) z^2 \tag{57}$$

$$\begin{aligned} u_3\ (r,z) = {} & \frac{556\gamma^2+1737\gamma+3069}{10{,}368} r^6 - \frac{388\gamma^2 + (1161-384\eta)\,\gamma + (1881-1728\eta)}{2304} r^4 \\ & + \frac{304\gamma^2 + (831-576\eta)\,\gamma + (1242-2160\eta+864\eta^2)}{1728} r^2 \\ & - \frac{2708\gamma^2 + (7839-5760\eta)\,\gamma + 14{,}211-32{,}832\eta) + 20{,}736\eta^2)}{82{,}944} \\ & + \left[\frac{52\gamma^2+51\gamma+327}{384} r^4 - \frac{52\gamma^2+(75\gamma^2+(279-288\eta)}{192} r^2 + \frac{92\gamma^2+180\gamma+(639-1080\eta+432\eta^2)}{1152}\right] z \\ & + \left[-\frac{7\gamma-3}{8} r^2 + \frac{(13-16\eta)\gamma-(27-24\eta)}{48}\right] r^2 + \left[\frac{4\gamma^2-57\gamma+27}{144}\right] r^3 \end{aligned} \tag{58}$$

$$v_1\ (r,z) = \tfrac{1}{4} r^3 - \tfrac{1}{4} r + rz \tag{59}$$

$$\begin{aligned} v_2\ (r,z) = {} & \frac{\gamma+3}{9} r^5 - \frac{20\gamma+63-36\eta}{96} r^3 + \frac{28\gamma+93-108\eta}{288} r + \left(\frac{2\gamma+9}{6} r^3 - \frac{4\gamma+15-12\eta}{12} r\right) z \\ & + rz^2 \end{aligned} \tag{60}$$

$$\begin{aligned} v_3\ (r,z) = {} & \frac{6836\gamma^2+23{,}031\gamma+30{,}627}{82{,}944} r^7 \ \frac{3380\gamma^2+(11{,}391-3840\eta)\gamma+(15{,}291-11{,}520\eta)}{13{,}824} r^5 \\ & + \frac{3424\gamma^2 + (11{,}271-7200\eta)\gamma+(15{,}228-22{,}680\eta+6480\eta^2)}{13{,}824} r^3 \\ & - \frac{7100\gamma^2 + (22{,}311-20{,}160\eta)\gamma + (30{,}249-66{,}960\eta+38{,}880\eta^2)}{82{,}944} r \\ & + \Bigg[\frac{556\gamma^2+1737\gamma+3069}{1728} r^5 - \frac{388\gamma^2+(1161-384\eta)\gamma+(1881-1728\eta)}{576} r^3 \\ & \qquad + \frac{304\gamma^2+(831-576\eta)\gamma+(1242-2160\eta + 864\eta^2)}{864} r\Bigg] z \\ & + \left[\frac{52\gamma^2+51\gamma+327}{192} r^3 - \frac{52\gamma^2+75\gamma+(279-288\eta)}{192} r\right] z^2 + \left[-\frac{7\gamma-3}{12} r\right] z^3 \end{aligned} \tag{61}$$

$$M^*\ (r,z) = 1 + u_1\varepsilon + u_2\varepsilon^2 + \left(u_3 + \frac{\gamma+1}{4} v_1^2\right)\varepsilon^3 + \ldots \tag{62}$$

$$\theta\ (r,z) = \left[\frac{\gamma+1}{2}\ \varepsilon\right]^{\frac{1}{2}} \left[v_1\ \varepsilon + (v_2-u_1v_1)\ \varepsilon^2 + (v_3-u_1v_2-u_2v_1+u_1^2v_1)\ \varepsilon^3 + \ldots\right] \quad (63)$$

This result is used by assuming that u>>v and the streamline deflection angle, Θ, is small then:

$$M^* \simeq u \quad (64)$$

$$\Theta \simeq v \quad (65)$$

since v is non-dimensionalized by a* and in the region of the throat $a^* \simeq M^*$. Θ is given by tan Θ = v/u and for small Θ, $\Theta \simeq = v/u$. The calculation procedure begins with the user choosing a value for R. The position on the axis where conical wall joins the circular throat can be calculated from:

$$X_c - R\sin\Theta_c \quad (66)$$

where Θ_c is the conical divergence angle of the nozzle and X_c is non-dimensionalized by r*. The distance from the axis to the wall, Y, at any axis position, X, is given by:

$$Y = 1 + R\ (1 - \sqrt{1 - \frac{X}{R}^{\ 2}} \quad (67)$$

Figure 3 illustrates this relationship. The velocity components as given by (54) and (55) are evaluated at increasing X until the axis velocity is slightly supersonic. At this point, values of u and v, hence M* and Θ, are calculated from the axis to the nozzle wall.

The values of M* and Θ along with the X and Y positions become the initial values for a method of characteristics routine which solves the flow in the nozzle until the local Mach number at the nozzle wall matches the model exit Mach number determined from the modeling theory. Then the X position at this point is the model nozzle length.

V. PLUME MODELING EXPERIMENTS

The Aeronautical Research Institute of Sweden (FFA) has developed a hot gas system for the FFA .5 x .5 meter supersonic wind tunnel for the purpose of studying plume effects. The efforts of FFA have been closely coordinated with members of the Gas Dynamics Laboratory at the University of Illinois at Urbana-Champaign.

Experiments were conducted by FFA with the expressed purpose of critically evaluating the merits and limitations of plume modeling techniques. Some of this effort has been documented by Nyberg et. al.[7]. For the plume modeling experiments reported in[7], FFA designed two air nozzles to be used as the prototype nozzles and two model Freon nozzles, one for weak shock modeling and one for strong shock modeling. The geometrical matching of plume shapes was verified by Schlieren photographs. Pressure measurements were made to verify the matching in the base region.

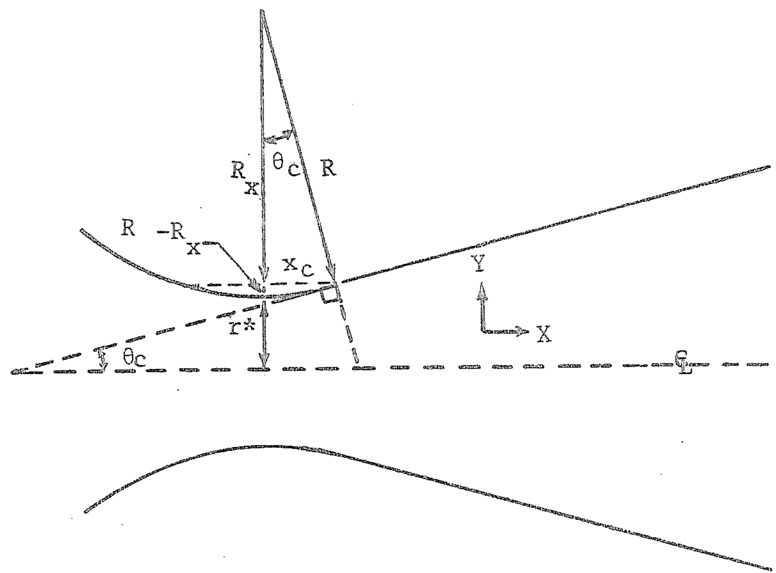

Figure 3. Nozzle geometry.

The matching of the plume shapes as presented in reference 7 is shown in Figures 4a and 4b. Figure 5, also from reference 7, demonstrates the base pressure matching where P_b is the base pressure, P_E is the exit pressure and P_L is the pressure at the nozzle lip. From these experiments, FFA concluded that the base pressure agreement was satisfactory at both the design point and for a wide range of off-design conditions. The overall conclusion of FFA was: "The Freon plumes shapes have been found to be in close agreement with those of the corresponding air test supporting the suggested modeling methodology and design precedures."[7] Testing by the Army, which will be documented at a later date tends to confirm this observation.

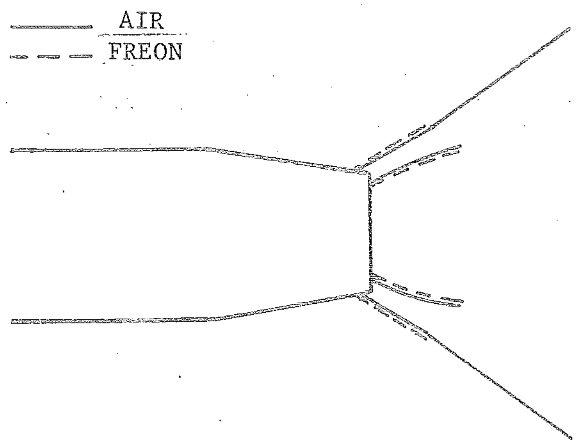

Figure 4a. Weak shock modeling.

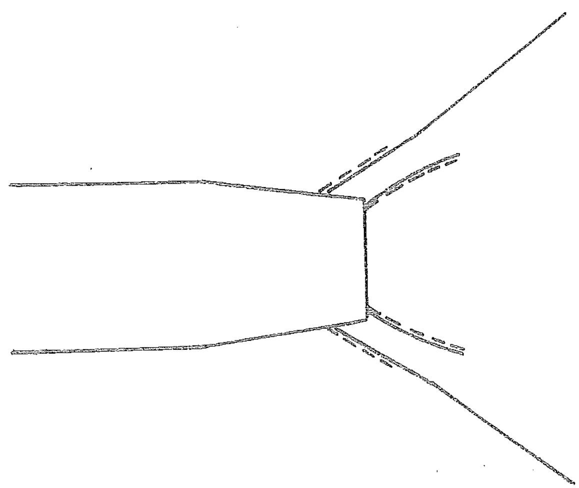

Figure 4b. Strong shock modeling.

Figure 4. Comparison of plume shape from Schlieren photos. $M_E = 2.0$; $\alpha = 0$. (Ref. 7).

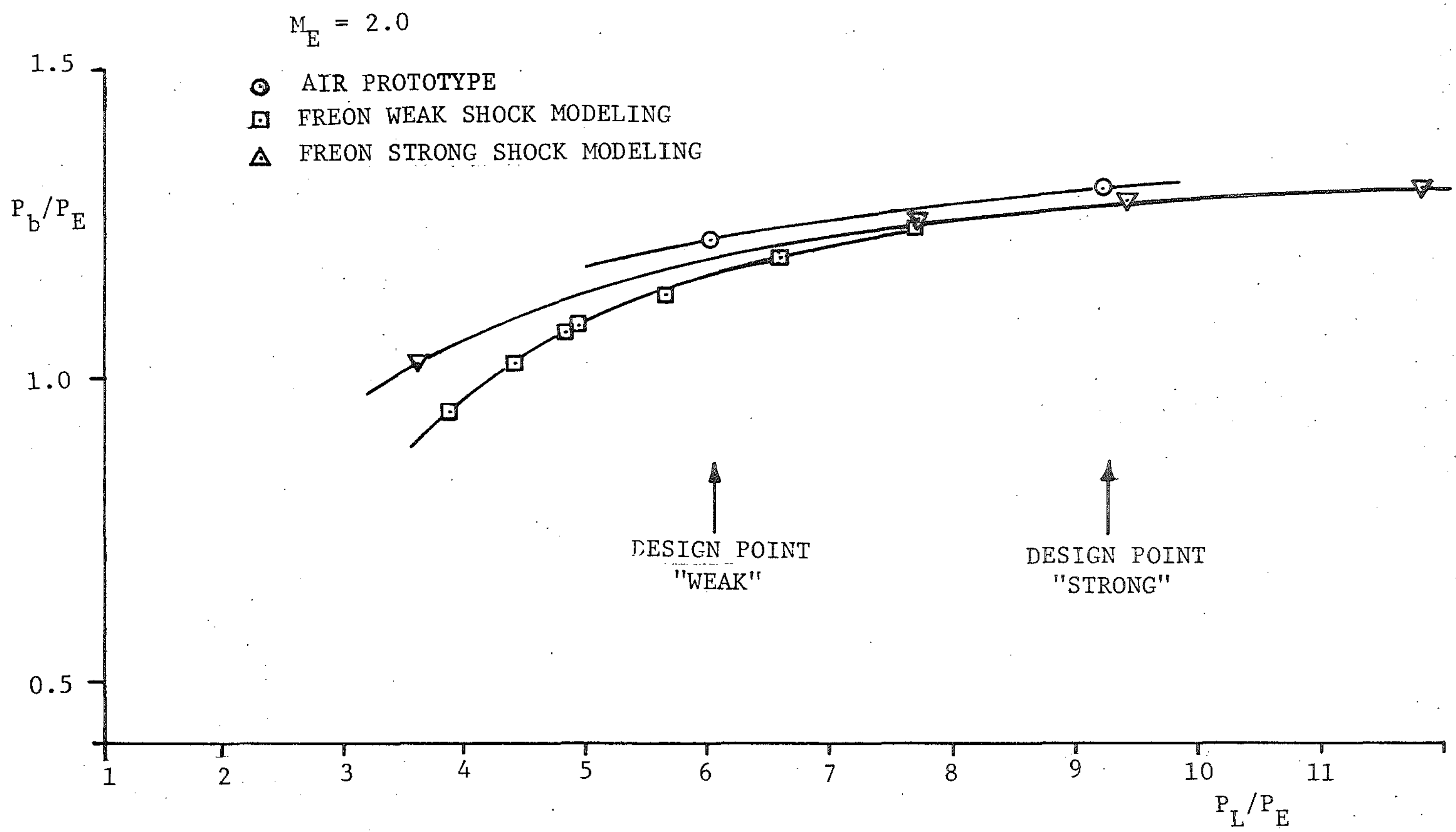

Figure 5. Base pressure versus lip pressure. Comparison of air prototype with freon models. (Ref. 7).

APPENDIX I

SAMPLE CASE

Tables I and II are the ordinary output from the model design program. Much more detailed nozzle flow field information and de-bugging information can be obtained by activating logical unit 6. There are some Perkin-Elmer Interdata 8/32 dependent statements in the program. For instance, subroutine PRTOPT which assigns the logical units for output and function IGC which is part of the free formatted read subroutine would have to be re-written for use on any other computer. However, the bulk of the program is written in standard Fortran IV.

The program is currently written to accept inputs interactively from a terminal. The prototype inputs for the sample case were: specific heat ratio, nozzle angle, exit Mach number and jet surface Mach number. Model inputs were: specific heat ratio and throat radius of curvature. All the other parameters shown in Table I. were calculated by the program. The weak shock modeling condition was chosen for the sample case.

Table II. tabulates the calculated plume shapes for the model and prototype. The prototype plume shape was calculated using the Johannesen-Meyer theory and the model plume shape was calculated using a method of characteristics routine which is started with exit conditions calculated from a separate nozzle method of characteristics routine. It is seen from Figure 6. that there is excellent agreement between the two plume shapes for approximately one body diameter.

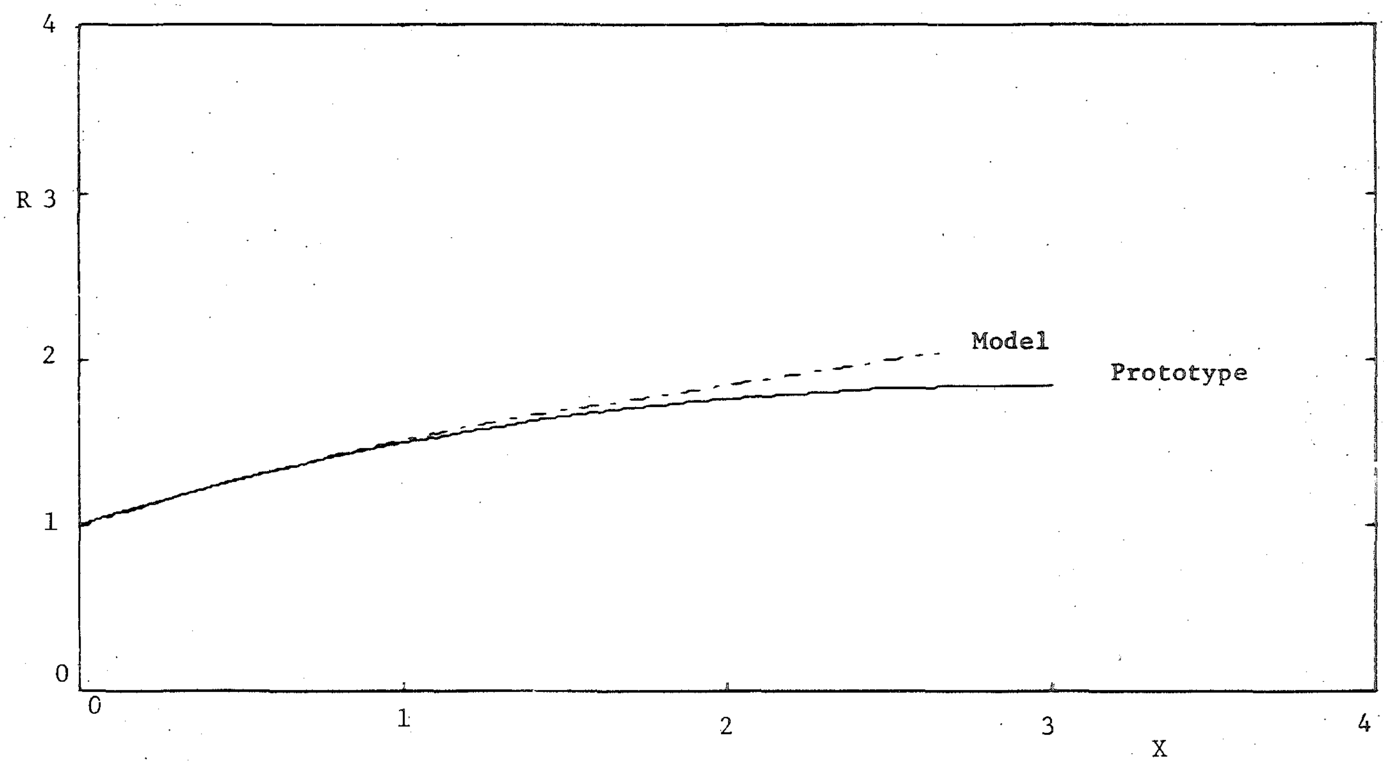

Figure 6. Comparison of prototype plume shape calculated by Johannesen - Meyer theory and mosel plume shape calculated by method of characteristics.

MODEL NOZZLE DESIGN

PROTOTYPE

SPECIFIC HEAT RATIO	1.235
NOZZLE ANGLE	10.360
EXIT MACH NUMBER	2.530
JET SURFACE MACH NUMBER	3.350
INITIAL SLOPE OF JET PLUME	32.304
INITIAL RADIUS OF CURVATURE OF JET PLUME	5.426
PRESSURE RATIO	0.012

MODEL

SPECIFIC HEAT RATIO	1.400
THROAT RADIUS OF CURVATURE	3.000
BEGINNING AXIAL LOCATION OF CONICAL SECTION	0.207
BEGINNING RADIAL LOCATION OF CONICAL SECTION	1.007
NOZZLE ANGLE, DEG	3.964
NOZZLE LENGTH	2.952
NOZZLE EXIT RADIUS	1.197
EXIT MACH NUMBER	1.761
JET SURFACE MACH NUMBER	2.907

TABLE I. MODEL AND PROTOTYPE CHARACTERISTICS.

PLUME SHAPE CALCULATED FROM JOHANNESEN-MEYER THEORY

X	R	THETA
0.0000	1.0000	32.3042
0.1000	1.0617	31.0633
0.2000	1.1205	29.8384
0.3000	1.1765	28.6283
0.4000	1.2297	27.432
0.5000	1.2803	26.2486
0.6000	1.3283	25.0771
0.7000	1.3739	23.9167
0.8000	1.4170	22.7667
0.9000	1.4578	21.6262
1.0000	1.4964	20.4947
1.1000	1.5326	19.3714
1.2000	1.5667	18.2559
1.3000	1.5986	17.1474
1.4000	1.6284	16.0456
1.5000	1.6561	14.9498
1.6000	1.6818	13.8596
1.7000	1.7055	12.7744
1.8000	1.7272	11.6939
1.9000	1.7469	10.6176
2.0000	1.7647	9.5451
2.1000	1.7805	8.4760
2.2000	1.7945	7.4098
2.3000	1.8065	6.3462
2.4000	1.8167	5.2848
2.5000	1.8250	4.2252
2.6000	1.8315	3.1670
2.7000	1.8361	2.1099
2.8000	1.8389	1.0536
2.9000	1.8398	-0.0024
3.0000	1.8388	-1.0585

MODEL PLUME SHAPE CALCULATED BY METHOD OF CHARACTERISTICS

X	R	THETA
0.0000	1.0000	32.3036
0.4153	1.2387	27.5741
0.6224	1.3430	25.7525
0.8406	1.4446	24.0213
1.0653	1.5411	22.4039
1.0654	1.5412	22.4048
1.3355	1.6478	20.5551
2.6375	2.0424	13.1913
2.6490	2.0451	13.0891

TABLE II. MODEL AND PROTOTYPE PLUME SHAPES

APPENDIX II.

MODEL NOZZLE DESIGN PROGRAM LISTING

```
C******************************************************************
C
C                        MODEL DESIGN PROGRAM
C
C******************************************************************
C
      REAL MLP,MFP,MFM,MLM
      INTEGER Z,Z0,S,P,Q,A8
      DIMENSION XA(3,45),R(3,45),XM(3,45),T(3,45),
     *HOL(20),FLO(20),B(45),C(45)
      COMMON/CKLEV/B,C
      COMMON/SHAPE/X7,C0,R7,T0,R0,GAMM,MLM
      COMMON/MARRAY/XA,XM,R,T
      COMMON/INDICE/I,N,K,IS,IP
      COMMON/INJET/MEXIT,MFM,K0,N0
      COMMON/VAR/VAR(17)
C*********************************************************************
C
5     CALL PRTOPT
C
C*********************************************************************
C*********************************************************************
C
      CALL MODEL(MFM,MEXIT,THTALM)
C
C*********************************************************************
C*********************************************************************
C
      CALL NOZZLE(MEXIT,THTALM)
C
C*********************************************************************
C*********************************************************************
C
      CALL PLUME
C
C*********************************************************************
C*********************************************************************
C
      CALL PLULIS
C
C*********************************************************************
C*********************************************************************
C
C     END OPTIONS
C
C*********************************************************************
      WRITE(1,130)
130   FORMAT(/T20'ENTER:'
     *T27'(1) RUN ANOTHER CASE'/
     *T27'(2) STOP')
      CALL FFREAD(HOL,FLO,LH,LF)
      IF(FLO(1).EQ.1.) GO TO 5
C
      STOP
      END
```

```
      SUBROUTINE MODEL(MFM,MEXIT,THTALM)
C
C  CONTROLS THE MODELING SECTION OF THE PROGRAM
C
      REAL MLP,MFP,MFM,MLM
      INTEGER Z,Z0,S,P,Q,A8
      DIMENSION XA(3,45),R(3,45),XM(3,45),T(3,45),HOL(20),FLO(20),
     *B(45),C(45)
      COMMON/CKLEV/B,C
      COMMON/SHAPE/X7,C0,R7,T0,R0,GAMM,MLM
      COMMON/MARRAY/XA,XM,R,T
C
      COMMON/RADS/PI,RAD
C
      PI=3.141592654
      RAD=57.3
1     IERR=0
C
C  SPECIFY TYPE OF INPUT
C
      WRITE(1,10)
10    FORMAT(/T3'ENTER:'
     */T5'1 PROTOTYPE INITIAL JET SLOPE AND RADIUS OF CURVATURE'
     *' KNOWN'
     */T5'2 PROTOTYPE NOZZLE GEOMETRY AND EXIT CONDITIONS KNOWN')
C
      CALL FFREAD(HOL,FLO,LH,LF)
      INOPT=FLO(1)
C
      GO TO (20,30),INOPT
C
C  PLUME SHAPE TO BE SPECIFIED
C
20    CALL PLUMIN(THTAFP,RADP)
      GO TO 40
C
C  PLUME FLOW CHARACTERISTICS TO BE SPECIFIED
C
30     CALL FLOWIN(GAMP,MLP,THTALP,MFP)
C
C  SPECIFY MODEL FLOW CONDITIONS
C
40    CALL MODIN(GAMM,MFM,GAMP,MFP,IERR)
      IF(IERR.EQ.1)GO TO 1
C
C  CALCULATE PLUME SHAPE IF PROTOTYPE FLOW SPECIFIED
C  (INOPT=2) AND PRINT RESULTS
C
      IF(INOPT.EQ.1)GO TO 50
      CALL CALCPL(GAMP,MLP,THTALP,MFP,THTAFP,RADP)
      CALL LISTPR(MFP,GAMP)
C
C  BEGIN ITERATION
C
50    CALL ITERAT(GAMM,THTAFP,THTALM,RADP,MFP,MFM,MLP,MLM)
C
```

```
RETURN
END
```

```
      SUBROUTINE NOZZLE(MEXIT,THTALM)
C
C  CONTROLS THE SECTION OF PROGRAM WHICH CALLS NOZZLE SOLUTION
C  AND USES METHOD OF CHARACTERISTICS TO CALCULATE THE NOZZLE
C  LENGTH.
C
      REAL MLP,MFP,MFM,MLM
      INTEGER Z,Z0,S,P,Q,A8
      DIMENSION XA(3,45),R(3,45),XM(3,45),T(3,45)
     *,B(45),C(45),HOL(20),FLO(20),QA(45),PA(45)
      COMMON/CKLEV/B,C
      COMMON/SHAPE/X7,C0,R7,T0,R0,G0,XM0
      COMMON/MARRAY/XA,XM,R,T
      COMMON/VAR/VAR(17)
      T0=THTALM*57.3
      K=9
C
      MEXIT=1
      VAR(8)=G0
      VAR(15)=XM0
      VAR(12)=T0
      T0=T0/57.3
74    WRITE(1,75)
75    FORMAT(/T20'ENTER THROAT RADIUS OF CURVATURE')
      CALL FFREAD(HOL,FLO,LH,LF)
      C0=FLO(1)
      VAR(9)=C0
C
      WRITE(1,138)
138   FORMAT(/T20'ENTER: 1-NOZZLE SOLUTION WITH KLIEGEL-LEVINE'
     1,/T27'2-NOZZLE SOLUTION WITH ADDY-DUTTON')
      CALL FFREAD(HOL,FLO,LH,LF)
      ISUB=FLO(1)
      X7=C0*SIN(T0)
      R7=1.+C0*(1.-SQRT(1.-(X7/C0)**2))
      VAR(10)=X7
      VAR(11)=R7
C
      WRITE(6,300)K
300   FORMAT(/1X,'NUMBER OF INITIAL POINTS ='I3)
C
C  CALL TO NOZZLE SUBROUTINES TO SOLVE THE FLOW FIELD
C  CLOSE TO THROAT  CONTINUES UNTIL X POSITION REACHED WHERE
C  MACH GREATER THAN 1.025.
C
C
      DO 90 X=0.0,X7,(X7/10.)-.000001
      Y0=1.+C0*(1.-SQRT(1.-(X/C0)**2))
C
      J=1
      DO 100 Y=0,Y0,(Y0/FLOAT(K-1))
      IF(ISUB.EQ.1)CALL KLGLEV(X,Y,J)
      IF(ISUB.EQ.2)CALL ADDUT(X,Y,J)
      PA(J)=X
      QA(J)=Y
      WRITE(6,400)X,Y
```

```
400   FORMAT(1X,'X = ',F20.8,' Y = ',F20.8)
      J=J+1
100   CONTINUE
      IF(B(1).GT.1.025)GO TO 32
90    CONTINUE
      WRITE(1,401)
401   FORMAT(/T5'** SUPERSONIC FLOW NOT ACHIEVED   INCREASE THROAT'
     *' RADIUS OF CURVATURE **')
      WRITE(1,402)B(1)
402   FORMAT(/T5'HIGHEST MACH BEFORE CONICAL SECTION = 'F8.4)
      GO TO 74
C
C  SETS UP INITIAL CONDITIONS FOR THE METHOD OF CHARACTERISTICS
C  PROGRAM WHERE XA ARE THE X LOCATIONS (ALL ARE EQUAL INITIALLY)
C                R  ARE THE RADIAL LOCATIONS
C                XM ARE THE DIMENSIONLESS SPEED MACH NUMBERS M*
C                T  ARE THE THETAS (ANGLE BETWEEN THE
C                   X CO-ORDINATE AND THE VELOCITY VECTOR.
C
32    DO 110 J=K,1,-1
      XA(1,K+1-J)=PA(J)
      R(1,K+1-J)=QA(J)
      XM(1,K+1-J)=B(J)
      T(1,K+1-J)=C(J)
110   CONTINUE
C
C  INITIALIZE POINTERS USED IN THE METHOD OF CHARACTERISTICS
C
      I=1
      S=1
      P=1
      A8=2
      Q=0
      N=0
      Z0=0
      K0=K
      IFINI=0
C
      DO 304 J=1,K
      WRITE(6,303)J,XA(I,J),R(I,J),XM(I,J),T(I,J)
303   FORMAT(1X,I3,2X,4F8.3)
304   CONTINUE
C
      WRITE(6,305)
305   FORMAT(/,5X,'FLOWFIELD COMPUTED  ALL POINTS PRINTED')
C
C  BEGIN METHOD OF CHARACTERISTICS LOOP
C
1260  CALL SETPTS(K,S,I)
      IF(I.NE.2)GO TO 1320
C
      CALL WALL(I)
      CALL CNTLIN(K,S,I)
      CALL YINCR(P,MEXIT,I,K,S)
1320  CALL SETEXT(P,A8,I,Q,K,S,N,IFINI)
      IF(IFINI.EQ.1)GO TO 140
```

```
C
      ITEST=(I/2)*2
      A8=1
      IF(ITEST.NE.I)A8=2
      CALL CROSS(I,K,S,A8)
C
      I=I+1
      S=S+1
C
      IF(I.NE.3)GO TO 1260
      CALL RESET(K,S)
      I=1
      GO TO 1260
140   RETURN
      END
```

```
      SUBROUTINE PLUME
C
C  CONTROLS THE SECTION OF THE PROGRAM WHICH CALCULATES
C  THE MODEL PLUME SHAPE  USING METHOD OF CHARACTERISTICS
C
      REAL MLP,MFP,MFM,MLM
      INTEGER Z,ZO,S,P,Q,A8
      DIMENSION XA(3,45),R(3,45),XM(3,45),T(3,45)
     *,B(45),C(45)
      COMMON/CKLEV/B,C
      COMMON/SHAPE/X7,CO,R7,TO,RO,GAMM,MLM
      COMMON/MARRAY/XA,XM,R,T
      COMMON/INDICE/I,N,K,IS,IP
      COMMON/INJET/MEXIT,MFM,KO,NO
      COMMON/VAR/VAR(17)
C
C  CALCULATE THE ACTUAL PLUME SHAPE USING METHOD OF CHARACTERISTICS
C
      NO=9
      KO=NO
      K=KO
      N=NO
      CALL JINIT
      I=1
270   CALL SETPT2
      CALL WRITRW
      I=I+1
      IS=IS+1
      IF(I.GT.(N-1))GO TO 1210
      GO TO 270
C
1210  J=1
1215  CALL PLUMPT
      CALL SETPT3
      N=N+1
      IP=IP+1
C
C  CHECK FOR END CONDITION
C
      IF(N.GT.(NO+KO-2))RETURN
      K=K-1
      GO TO 1215
C
      RETURN
      END
```

```
      SUBROUTINE ADDUT(X,Y,J)
      REAL N,N2
      DIMENSION B(45),C(45)
      COMMON/CKLEV/B,C
      COMMON/SHAPE/X7,CO,R7,TO,RO,G,XMO
C
      N=1.
      E=1./(CO+N)
      Z = X/SQRT(((G+1.)/2.)*E)
      Y2=Y*Y
      Y3=Y2*Y
      Y4=Y3*Y
      Y5=Y4*Y
      Y6=Y5*Y
      Y7=Y6*Y
C
      E2=E*E
      E3=E2*E
C
      Z2=Z*Z
      Z3=Z2*Z
C
      G2=G*G
C
      N2=N*N
C
      U1=.5*Y2-.25+Z
      V1=(.25*Y2-.25+Z)*Y
      U2=((2.*G+9.)/24.)*Y4 - ((4.*G+15.-12.*N)/24.)*Y2
     A + (10.*G+57.-72.*N)/288. + (Y2+(4.*N-5.)/8.)*Z
     B - ((2.*G-3.)/6.)*Z2
C
      V2=((G+3.)/9.)*Y5 - ((20.*G+63.-36.*N)/96.)*Y3
     A + ((28.*G+93.-108.*N)/288.)*Y
     B +(((2.*G+9.)/6.)*Y3 - ((4.*G+15.-12.*N)/12.)*Y)*Z+Y*Z2
C
      U3=((556.*G2+1737.*G+3069.)/10368.)*Y6
     A -((388.*G2+(1161.-384.*N)*G+(1881.-1728.*N))/2304.)*Y4
     B+((304.*G2+(831.-576.*N)*G+(1242.-2160.*N+864.*N2))/1728.)*Y2
     C-(2708.*G2+(7839.-5760.*N)*G+(14211.-32832.*N+20736.*N2))/82944.
     D+((52.*G2+51.*G+327.)/384.)*Y4*Z
     E-((52.*G2+75.*G+(279.-288.*N))/192.)*Y2*Z
     F+((92.*G2+180.*G+(639.-1080.*N+432.*N2))/1152.)*Z
     G-((7.*G-3.)/8.)*Y2*Z2
     H+(((13.-16.*N)*G-(27.-24.*N))/48.)*Z2
     I+((4.*G2-57.*G+27.)/144.)*Z3
C
      V3=((6836.*G2+23031.*G+30627.)/82944.)*Y7
     A-((3380.*G2+(11391.-3840.*N)*G+(15291.-11520.*N))/13824.)*Y5
     B+((3424.*G2+(11271.-7200.*N)*G+(15228.-22680.*N+6480.*N2)
     1 )/13824.)*Y3
     C-((7100.*G2+(22311.-20160.*N)*G+(30249.-66960.*N+38880.*N2)
     1 )/82944.)*Y
     D+((556.*G2+1737.*G+3069.)/1728.)*Y5*Z
     E-((388.*G2+(1161.-384.*N)*G+(1881.-1728.*N))/576.)*Y3*Z
     F+((304.*G2+(831.-576.*N)*G+(1242.-2160.*N+864.*N2))/864.)*Y*Z
```

```
      G+((52.*G2+51.*G+327.)/192.)*Y3*Z2
      H-((52.*G2+75.*G+(279.-288.*N))/192.)*Y*Z2
      I-((7.*G-3.)/12.)*Y*Z3
C
       B(J)=1.+U1*E+U2*E2+((U3+(G+1.)/.25)*V1*V1)*E3
C
       C(J)=SQRT((G+1.)/2.*E)*(V1*E+(V2-U1*V1)*E2
      A +(V3-U1*V2-U2*V1+U1*U1*V1)*E3)
C
C
       RETURN
       END
```

```
      SUBROUTINE CALCPL(GAMMA,MACHL,THTAL,MACHF,THTAF,RADIUS)
      REAL LAMDA,MUL,MACHL,MACHF,MUF
      COMMON/RADS/PI,RAD
C
C  THIS ROUTINE CALCULATES THE INITIAL SLOPE OF THE PLUME
C  AND THE INITIAL RADIUS OF CURVATURE OF THE PLUME.
C  THE FLOW OUTSIDE OF THE CENTERED WAVE REGION IS CALCULATED
C  BY ESTABLISHING BOUNDARY CONDITIONS FROM MATCHING
C  SOLUTIONS AT THE LOWER (PHIL) AND FINAL (PHIF) MACH LINES
C  OF THE CENTERED FAN REGION.
C
C
      LAMDA=SQRT((GAMMA-1.)/(GAMMA+1.))
      MUL=XMANG1(1./MACHL)
      PHIL=THTAL-MUL+PI/2.
      E1=ATAN(LAMDA/TAN(MUL))
      BO=E1/LAMDA-PHIL
C
C  X1 AND X2 ARE THE M* MACH VALUES X1=M*L AND X2=M*F
C
      X1=XMSQR(MACHL,GAMMA)
      X2=XMSQR(MACHF,GAMMA)
C
C  CALCULATES THE PRANDTL-MAYER FUNCTIONS FOR M*L AND M*F
C  XL1=W(M*L) AND XL2=W(M*F).  THEN XL2-XL1 IS THE
C  DELTA TURNING ANGLE DUE TO THE LIP.
C
      CALL TANGLE(X1,XL1,LAMDA)
      CALL TANGLE(X2,XL2,LAMDA)
C
C THE FOLLOWING EQUATIONS CALCULATE THE APPROACHING FLOW
C  IN REGION B (OUTSIDE OF THE CENTERED WAVE)
C
      CO=SIN(THTAL)*(4.*(COS(MUL))**2/
     *(MACHL*MACHL-1.)-(SIN(MUL))**2)
C
      C2=-2.*LAMDA*SQRT(X1)/(COS(E1)**(((3.*GAMMA)-1.)/(2.*(GAMMA-1.)))
     1*SIN(E1)**.5)
C
      C1=CO*C2
      E2=E1+LAMDA*(XL2-XL1-XMANG1(1./MACHF)+XMANG1(1./MACHL))
C
C  THE FOLLOWING SECTION INTEGRATES THE INTEGRALS OCCURING
C  IN THE CENTERED WAVE EQUATIONS.
C
      D1=(E2-E1)/10.
C
      S1=0.0
      S2=0.0
      S3=0.0
      S4=0.0
C
      DO 10 XN=1.,10.
      E3=E1+D1*(XN-.5)
      S1=COS(E3/LAMDA)*SIN(E3)**(-.5)*COS(E3)**
     *(-1./(2.*LAMDA*LAMDA))*D1+S1
```

```
C
      S2=SIN(E3/LAMDA)*SIN(E3)**(-.5)*COS(E3)**
     *(-1./(2.*LAMDA*LAMDA))*D1+S2
C
      S3=SIN(E3/LAMDA)*SIN(E3)**(-1.5)*COS(E3)**
     *((GAMMA-3.)/(2.*(GAMMA-1.)))*D1+S3
C
      S4=COS(E3/LAMDA)*SIN(E3)**(-1.5)*COS(E3)**
     *((GAMMA-3.)/(2.*(GAMMA-1.)))*D1+S4
C
10    CONTINUE
C
      U0=-((COS(E2)**((3.*GAMMA-1.)/(2.*(GAMMA
     *-1.))))*SIN(E2)**.5)/(2.*LAMDA)
C
C
C  U1 IS THE VELOCITY AT THE UPPER LIMIT OF THE CENTERED WAVE
C  REGION.
C
      U1=U0*((S1-LAMDA*S3)*COS(B0)+(S2+LAMDA*S4)*
     *SIN(B0)+C1)
C
      MUF=XMANG1(1./MACHF)
      THTAF=THTAL+XL2-XL1
C
      R0=-((U1/SQRT(X2)+SIN(THTAF)*SIN(MUF)**2)
     */(SIN(2.*MUF)))
C
      RADIUS=-1./R0
C
      RETURN
      END
```

```
      SUBROUTINE CNTLIN(K,S,I)
C
C  CALCULATES CONDITIONS AT THE CENTERLINE BOUNDARY CONDITION
C
      REAL M(3,45)
      INTEGER S
      DIMENSION X(3,45),T(3,45),R(3,45)
      COMMON/MARRAY/X,M,R,T
      COMMON/LASTV/XML,TL,AL
      COMMON/SHAPE/X7,CO,R7,TO,RO,GO,XMO
C
      WRITE(6,7)
7     FORMAT(/T20'***  CNTLIN  ***')
C
      X1=X(I,K-S+1)
      R1=R(I,K-S+1)
      XM1=M(I,K-S+1)
      T1=T(I,K-S+1)
      A1=XMANG2(XMLOC(XM1,GO))
      XM9=1.01
      A4=A1
      T4=T1/2.
      XM4=XM1
      R4=R1/2.
2760  T9=TAN(A4-T4)
      X3=X1+R1/T9
      G1=FNG(A4,TL)
      Q1=FNQ(A4,XML)
C
C  COMPATIBILITY EQUATION FOR THE CENTERLINE
C
      XM3=XM1+(T1-G1)/Q1
      IF(ABS(XM3-XM9).LT. .0001)GO TO 2870
C
C
C
      XM9=XM3
      XM4=(XM1+XM3)/2.
      A3=XMANG2(XMLOC(XM3,GO))
      A4=(A1+A3)/2.
      GO TO 2760
C
C  AT CENTERLINE THE RADIAL CO-ORDINATE=0 AND THETA=0
C
2870  X(I+1,K-S+1)=X3
      R(I+1,K-S+1)=0.0
      M(I+1,K-S+1)=XM3
      T(I+1,K-S+1)=0.0
C
      RETURN
      END
C
C
C
C
C
```

```
      SUBROUTINE CROSPT(I,J)
      REAL M(3,45)
      DIMENSION R(3,45),T(3,45),X(3,45),XH(5),XI(5),XK(5),XJ(5)
C
      COMMON/MARRAY/X,M,R,T
      COMMON/CNETPT/X3,R3,XM3,T3
      COMMON/CCROSS/XH,XI,XJ,XK
      COMMON/CBKPTS/X1,X2,R1,R2,XM1,XM2,T1,T2
      WRITE(6,7)
7     FORMAT(/T20'***  CROSPT  ***')
C
C  CHARACTERISTICS HAVE CROSSED  INTERPOLATE TO DETERMINE
C  A NEW POINT.
C
      S8=(XH(2)-XH(1))/(XK(2)-XK(1))
      S9=(XH(4)-XH(3))/(XK(4)-XK(3))
      S6=(XH(3)-XH(1)+XK(1)*S8-XK(3)*S9)/(S8-S9)
      S7=XH(1)+S8*(S6-XK(1))
C
      WRITE(6,10)S6,S7
10    FORMAT(/T20,'X(C)=',F10.4,2X,'R(C)=',F10.4)
C
      V1=XI(1)+(XI(2)-XI(1))*(S6-XK(1))/(XK(2)-XK(1))
      V2=XI(3)+(XI(4)-XI(3))*(S6-XK(3))/(XK(4)-XK(3))
      V3=XJ(1)+(XJ(2)-XJ(1))*(S6-XK(1))/(XK(2)-XK(1))
      V4=XJ(3)+(XJ(4)-XJ(3))*(S6-XK(3))/(XK(4)-XK(3))
C
      X1=S6
      R1=S7
      XM1=(V1+V2)/2.
      T1=(V3+V4)/2.
      IF(XH(2).EQ.0.0)GO TO 4100
      X2=XK(5)
      R2=XH(5)
      XM2=XI(5)
      T2=XJ(5)
C
      CALL NETPT
      GO TO 4150
C
4100  WRITE(6,20)
20    FORMAT(/T20'NEW POINT IS ON AXIS')
C
C  ITERATE IF POINT ON CENTERLINE
C
      R3=0.0
      T3=0.0
      A1=XMANG2(XMLOC(XM1,GO))
      XM9=1.01
      A4=A1
      T4=T1/2.
      XM4=XM1
      R4=R1/2.
C
5540  T9=TAN(A4-T4)
      X3=X1+R1/T9
```

```
      G1=FNG(A4,T4)
      Q1=FNQ(A4,XM4)
      XM3=XM1+(T1-G1)/Q1
      IF(ABS(XM3-XM9).LT..0001)GO TO 4150
      XM9=XM3
      XM4=(XM1+XM3)/2.
      A3=XMANG2(XMLOC(XM3,G0))
      A4=(A1+A3)/2.
      GO TO 5540
C
C
4150  X(I,J)=X3
      R(I,J)=R3
      M(I,J)=XM3
      T(I,J)=T3
      T3D=T3*57.2958
C
      WRITE(6,30)X3,R3,XM3,T3D
30    FORMAT(/T10'X3='F10.4,2X,'R3='F10.4,2X,'M3='F10.4,
     *2X,'T3='F10.4)
      RETURN
      END
```

```
      SUBROUTINE CROSS(I,K,S,A8)
      INTEGER S,A8,P,P0,Z0,Z1
      REAL M(3,45)
      DIMENSION R(3,45),T(3,45),X(3,45),XI(5),
     *XJ(5),XK(5),XH(5)
      COMMON/MARRAY/X,M,R,T
      COMMON/CNETPT/X3,R3,XM3,T3
      COMMON/CCROSS/XH,XI,XJ,XK
      COMMON/CBKPTS/X1,X2,R1,R2,XM1,XM2,T1,T2
C
C  CHECKS IF CHARACTERISTICS OF THE SAME FAMILY HAVE CROSSED
C  IF CHARACTERISTICS HAVE CROSSED INTERPOLATE TO DETERMINE A
C  A NEW POINT.
C
C
      WRITE(6,7)
7     FORMAT(/T20'***  CROSS  ***')
      WRITE(6,10)I,K,S
10    FORMAT(/T20,'I='I3,2X'K='I3,2X'S='I3)
      I1=I+1
      IF(I.EQ.1)RETURN
      KS1=K-S+1
      IF(A8.EQ.1)KS1=K-S-1
C
      DO 20 J=2,KS1
C
      J1=J-1
      J2=J1
      J3=J
      J4=J-1
      IF(A8.NE.1)GO TO 5
      J2=J
      J3=J+1
      J4=J+1
C
5     IF(A8.EQ.2.AND.X(I,J).LT.X(I1,J1))GO TO 20
      IF(A8.EQ.1.AND.X(I,J).LT.X(I1,J))GO TO 20
C
      WRITE(6,30)
30    FORMAT(/T20'CHARACTERISTICS HAVE CROSSED')
      J0=J
      XK(1)=X(I-1,J2)
      XH(1)=R(I-1,J2)
      XI(1)=M(I-1,J2)
      XJ(1)=T(I-1,J2)
C
      XK(2)=X(I,J)
      XH(2)=R(I,J)
      XI(2)=M(I,J)
      XJ(2)=T(I,J)
C
      XK(3)=X(I,J1)
      XH(3)=R(I,J1)
      XI(3)=M(I,J1)
      XJ(3)=T(I,J1)
      XK(4)=X(I1,J2)
```

```
      XH(4)=R(I1,J2)
      XI(4)=M(I1,J2)
      XJ(4)=T(I1,J2)
C
      XK(5)=X(I-1,J3)
      XH(5)=R(I-1,J3)
      XI(5)=M(I-1,J3)
      XJ(5)=T(I-1,J3)
C
4710  WRITE(6,40)(XK(IJ),IJ=1,5)
40    FORMAT(1X,5F12.4)
      CALL CROSPT(I,J)
C
      X(I,J)=X3
      R(I,J)=R3
      M(I,J)=XM3
      T(I,J)=T3
C
      X1=X3
      R1=R3
      XM1=XM3
      T1=T3
      IF(XH(2).LT.0.0)GO TO 4710
      X2=X(I,J+1)
      R2=R(I,J+1)
      T3=T(I,J+1)
      IF(A8.EQ.2)XM2=M(I,J+1)
      IF(A8.EQ.1)XM3=M(I,J+1)
      CALL NETPT
C
      X(I1,J4)=X3
      R(I1,J4)=R3
      M(I1,J4)=XM3
      T(I1,J4)=T3
C
      WRITE(6,40)X3,R3,XM3,T3
      K=K-1
      KS1=K-S-1
      DO 60 P0=J0+1,KS1
      X(I1,P0-1)=X(I1,P0)
      R(I1,P0-1)=R(I1,P0)
      M(I1,P0-1)=M(I1,P0)
      T(I1,P0-1)=T(I1,P0)
60    CONTINUE
      DO 70 Z1=J0-1,K-S
      WRITE(6,80)I1,Z1,X(I1,Z1),R(I1,Z1),M(I1,Z1),T(I1,Z1)
80    FORMAT(/T30'CORRECTED POINTS',/,2I3,4F12.4)
70    CONTINUE
20    CONTINUE
      RETURN
      END
```

```
$PROG FFREAD
C     ************************************************************
      SUBROUTINE FFREAD (HOL,FLO,LH,LF)
C     ************************************************************
      INTEGER HOL
      LOGICAL TRANSP,DIGIT,COMENT,DELIO,DELTO5
      DIMENSION FORM(6),IBUF( 20),HOL(20),FLO(20)
      DATA NWORDS, NIT, NOT, NCHARS /  20, 5, 6,4 /
      TRANSP(K)=K.EQ.IBLANK.OR.K.EQ.ICOMMA.OR.K.EQ.IEQUAL
      DIGIT(K) =K.GE.IZERO.AND.K.LE.ININE
      COMENT(A)=IGC(A,1).EQ.ILETTC.AND.IGC(A,2).EQ.IBLANK
C-----------------------------------------------------------------
C
C PURPOSE
C -------
C
C READS ONE CARD IN FREE FIELD FORMAT
C THE $ SIGN MAY BE USED TO CANCEL ITEMS OR AN ENTIRE LINE
C ANY COMMENT CARDS ENCOUNTERED IN THE DATA DECK ARE
C PRINTED, SKIPPED AND NOT INTERPRETED
C
C ARGUMENTS  I=INPUT
C ---------  O=OUTPUT
C
C HOL   -- O -- WILL CONTAIN HOLLERITHS ENCOUNTERED ON CARD
C FLO   -- O -- WILL CONTAIN FLOATS ENCOUNTERED ON CARD
C INT   -- O -- WILL CONTAIN INTEGERS ENCOUNTERED ON CARD
C LH    -- O -- NUMBER OF HOLLERITHS ON CARD
C LF    -- O -- NUMBER OF FLOATS ON CARD
C LI    -- O -- NUMBER OF INTEGERS ON CARD
C
C DEFINITIONS -- LIMITATIONS
C --------------------------
C
C A *TRACAR* (TRANSPARENT CHARACTER) IS A BLANK, COMMA OR = SIGN
C THE FOLLOWING ARE DELIMITERS
C - BEGIN OF CARD
C - END OF CARD
C - A STRING CONSISTING OF ONE OR MORE TRACARS
C AN ITEM CONSISTS OF ONE OR MORE NON-TRACARS PRECEDED AND FOLLOWED BY A
C DELIMITER.
C EACH ITEM ON A CARD WILL BE INTERPRETED AS ONE OF THE FOLLOWING
C - INTEGER
C - FLOAT
C - HOLLERITH
C ANY ITEM STARTING WITH + - . OR DIGIT WILL BE INTERPRETED AS
C - INTEGER IF IT CONTAINS NO .
C - FLOAT IF IT CONTAINS ONE .
C - HOLLERITH IF IT CONTAINS MORE THAN ONE .
C ALL OTHER ITEMS WILL BE INTERPRETED AS HOLLERITHS
C GOOD INTEGERS -- 1 +5 -208 0034 -051
C BAD  INTEGERS -- ++5 -6+3 2A
C GOOD FLOATS   -- 1.E+5  2.-3  -6.5 +006.5 .2
C BAD  FLOATS   -- 1.A+5  2.-3X  +6.6--6
C HOLLERITHS    -- .. A 12..6 +3..4  XAQ X5 * A$A $$$$X
C CANCELED ITEMS-- A$ 43$ 12..6$ XXXX$ 1$ 12$ 123$ 1234$ $$ $$$
```

```
C CANCELED LINE -- 2.8 4.2 1264 SHIFT $
C
C-----------------------------------------------------------------
      IPLUS =IGC(1H+,1)
      ILETTC=IGC(1HC,1)
      IMIN  =IGC(1H-,1)
      IDOLL =IGC(1H$,1)
      IDOT  =IGC(1H.,1)
      IBLANK=IGC(1H ,1)
      ICOMMA=IGC(1H,,1)
      IEQUAL=IGC(1H=,1)
      IZERO =IGC(1H0,1)
      ININE =IGC(1H9,1)
C-----------------------------------------------------------------
C     INITIALIZE COUNTS AND READ CARD
C-----------------------------------------------------------------
 1020 LH=0
      LF=0
      LI=0
      READ (1,1060)(IBUF(I),I=1,NWORDS)
 1060 FORMAT (20A4)
      IF (COMENT(IBUF(1))) GO TO 1020
      I2=0
C-----------------------------------------------------------------
C     FIND I1 = FIRST COLUMN OF FIELD
C-----------------------------------------------------------------
 1100 I1=I2
 1120 I1=I1+1
      IF (I1.GT.80) RETURN
      IF=IGC(IBUF,I1)
      IF (TRANSP(IF)) GO TO 1120
C-----------------------------------------------------------------
C     FIND I2 = LAST COLUMN OF FIELD
C-----------------------------------------------------------------
      I2=I1
 1140 I2=I2+1
      IF (I2.GT.80) GO TO 1160
      IL=IGC(IBUF,I2)
      IF (.NOT.TRANSP(IL)) GO TO 1140
      I2=I2-1
      GO TO 1180
 1160 I2=80
C-----------------------------------------------------------------
C     IW = FIELD WIDTH
C-----------------------------------------------------------------
 1180 NX=I1-1
      IW=I2-I1+1
      NCH=NX+IW
      DO 1190 I=1,6
 1190 FORM(I) =4H
      IL=IGC(IBUF,I2)
      IF (IL.EQ.IDOLL.AND.I1.EQ.I2) GO TO 1020
      IF (IL.EQ.IDOLL) GO TO 1100
      IF=IGC(IBUF,I1)
      IF (IF.EQ.IPLUS) GO TO 1200
      IF (IF.EQ.IMIN) GO TO 1200
```

```
      IF (IF.EQ.IDOT) GO TO 1200
      IF (DIGIT(IF)) GO TO 1200
      GO TO 1240
C-------------------------------------------------------------
C     COUNT THE DOTS
C-------------------------------------------------------------
 1200 NDOTS=0
      DO 1220 I=I1,I2
      IF (IGC(IBUF,I).EQ.IDOT) NDOTS=NDOTS+1
 1220 CONTINUE
      IF (NDOTS.EQ.0) GO TO 1440
      IF (NDOTS.EQ.1) GO TO 1340
C-------------------------------------------------------------
C     HOLLERITH
C-------------------------------------------------------------
 1240 IW=MINO(IW,NCHARS)
      NCH=NX+IW
      IF (NX.EQ.0) GO TO 1280
      ENCODE (FORM,1260)NX,IW
 1260 FORMAT (  1H(,  I2,  4HX,1A,  I2,  1H)  )
      GO TO 1320
 1280 ENCODE (FORM,1300)IW
 1300 FORMAT (  3H(1A,  I2,  1H)  )
 1320 LH=LH+1
      DECODE (IBUF,FORM)HOL(LH)
      GO TO 1100
C-------------------------------------------------------------
C     FLOAT
C-------------------------------------------------------------
 1340 IF (NX.EQ.0) GO TO 1380
      ENCODE (FORM,1360)NX,IW
 1360 FORMAT (  1H(,  I2,  3HX,F,  I2,  3H.1)  )
      GO TO 1420
 1380 ENCODE (FORM,1400)IW
 1400 FORMAT (  2H(F,  I2,  3H.1)  )
 1420 LF=LF+1
      DECODE (IBUF,FORM)FLO(LF)
      GO TO 1100
C-------------------------------------------------------------
C     INTEGER
C-------------------------------------------------------------
 1440 IF (NX.EQ.0) GO TO 1480
      ENCODE (FORM,1460)NX,IW
 1460 FORMAT (  1H(,  I2,  3HX,I,  I2,  1H)  )
      GO TO 1520
 1480 ENCODE (FORM,1500)IW
 1500 FORMAT (  2H(I,  I2,  1H)  )
 1520 CONTINUE
      DECODE (IBUF,FORM)INT
      LF=LF+1
      FLO(LF)=INT
      GO TO 1100
      END
C     ***********************************************************
      FUNCTION IGC (ISTR,N)
C     ***********************************************************
```

```
      DIMENSION ISTR(1)
      DOUBLE PRECISION ISUB
      DATA ISUB,NCHARS / 3HIGC,4 /
C------------------------------------------------------------------
C       THIS FUNCTION RETRIEVES THE N-TH CHARACTER IN THE STRING
C <ISTR> AND STORES ITS OCTAL 6-BIT CODE IN THE 6 LEAST
C SIGNIFICANT BITS OF (IGC>.
C  NOTE THIS ROUTINE CONTAINS INTERDATA 8/32 DEPENDENT STATEMENTS
C------------------------------------------------------------------
      MWORD = Y'000000FF'
      ICHAR=N-1
      IWORD=ICHAR/NCHARS
      IRELA=ICHAR-IWORD*NCHARS
      IRELA=(NCHARS-1)-IRELA
      ISHIF=8*IRELA
      JWORD=ISTR(IWORD+1)
      JWORD=ISHFT(JWORD,-ISHIF)
      IGC   = IAND(JWORD,MWORD)
      RETURN
      END
```

```
      SUBROUTINE FLOWIN(GAMP,MLP,THTALP,MFP)
      REAL MLP,MFP
      DIMENSION HOL(20),FLO(20)
      COMMON/RADS/PI,RAD
      COMMON/VAR/VAR(17)
C
C  THIS SUBROUTINE ALLOWS INPUT OF PROTOTYPE
C  NOZZLE FLOW CONDITIONS
C
      WRITE(1,10)
10    FORMAT(/T20'PROTOTYPE FLOW SPECIFIED'
     *//T20'ENTER GAMMA OF PROTOTYPE')
      CALL FFREAD(HOL,FLO,LH,LF)
      GAMP=FLO(1)
C
      WRITE(1,20)
20    FORMAT(/T20'ENTER PROTOTYPE NOZZLE EXIT MACH')
      CALL FFREAD(HOL,FLO,LH,LF)
      MLP=FLO(1)
C
      WRITE(1,30)
30    FORMAT(/T20'ENTER PROTOTYPE NOZZLE EXIT ANGLE')
      CALL FFREAD(HOL,FLO,LH,LF)
      THTALP=FLO(1)
C
      WRITE(1,40)
40    FORMAT(/T20'ENTER PROTOTYPE JET SURFACE MACH NUMBER')
      CALL FFREAD(HOL,FLO,LH,LF)
      MFP=FLO(1)
C
      VAR(1)=GAMP
      VAR(2)=THTALP
      VAR(3)=MLP
      VAR(4)=MFP
C
      THTALP=THTALP/RAD
C
      RETURN
      END
```

```
      SUBROUTINE INTERM
      DIMENSION HOL(20),FLO(20),XI(11),XJ(11),XK(11),XH(11)
      COMMON/C1005/XH,XI,XJ,XK
1330  WRITE(1,10)
10    FORMAT(/T20'ENTER SEGMENT')
      CALL FFREAD(HOL,FLO,LH,LF)
      IP=FLO(1)+1
      WRITE(1,20)
20    FORMAT(/T20'ENTER RADIUS')
      CALL FFREAD(HOL,FLO,LH,LF)
      Q=FLO(1)
      U=XJ(IP-1)
      U1=COS(U)
      U2=U1+(Q-XI(IP-1))/XK(IP)
      U2=ATAN(SQRT(1.-U2*U2)/U2)
      HPKP=XH(IP-1)+XK(IP)*(SIN(U)-SIN(U2))
      U2D=U2*57.3
      WRITE(6,30)Q,U2D,HPKP
30    FORMAT(1X,' R=',F10.4,' THETA=',F10.4,' X=',F10.4)
      GOTO 1330
      END
```

```
      SUBROUTINE ITERAT(GAMM,THTAF,THTALM,RADIUS,MFP,MFM,MLP,MLM)
      REAL LAMDA,MFM,ML,ML2,ML3,MLP,MLM,MFP
      COMMON/RADS/PI,RAD
      COMMON/VAR/VAR(17)
C
C  THIS SUBROUTINE CALCULATES A VALUE FOR THETA-L-MODEL
C  BASED ON MODEL GAMMA AND A GUESS AT MACH-L-MODEL
C
C
      ML=MLP
C
      LAMDA=SQRT((GAMM-1.)/(GAMM+1.))
      Q=THTAF
      RAD2=RADIUS
      X1=XMSQR(ML,GAMM)
      X2=XMSQR(MFM,GAMM)
      CALL TANGLE(X1,XL1,LAMDA)
      CALL TANGLE(X2,XL2,LAMDA)
      THTALM=Q-XL2+XL1
      THDEG=THTALM*RAD
      WRITE(6,20)ML,THDEG
20    FORMAT(/T20'ML-M-1='F10.4,'THETA-L-M='F10.4)
C
      CALL CALCPL(GAMM,ML,THTALM,MFM,THTAF,RADIUS)
C
C  SECOND PASS
C
      ML2=ML
      RAD3=RADIUS
      ML=MLP+2.*(MFM-MFP)
C
      X1=XMSQR(ML,GAMM)
      CALL TANGLE(X1,XL1,LAMDA)
      THTALM=Q-XL2+XL1
      THDEG=THTALM*RAD
      WRITE(6,40)ML,THDEG
40    FORMAT(/T20'ML-M-2='F10.4,'THETA-L-M-2='F10.4)
      ML3=ML
C
      CALL CALCPL(GAMM,ML,THTALM,MFM,THTAF,RADIUS)
C
C
C  THIRD PASS
C
50    ML3=ML
      RAD4=RADIUS
      ML=ML+( RAD4-RAD2)*(ML3-ML2)/(RAD3-RAD4)
      IF(ML.LT.1.)ML=(ML2+1.)/2.
      X1=XMSQR(ML,GAMM)
      CALL TANGLE(X1,XL1,LAMDA)
      THTALM=Q-XL2+XL1
      THDEG=THTALM*RAD
      WRITE(6,60)ML,THDEG
60    FORMAT(/T20'ML-M='F10.4,2X,'THETA-L-M='F10.4)
C
C  TEST FOR CLOSURE
```

```
C
      IF(ABS(RAD2-RAD4).LT..01)GO TO 70
      ML2=ML3
      RAD3=RAD4
      CALL CALCPL(GAMM,ML,THTALM,MFM,THTAF,RADIUS)
      GO TO 50
C
C  END CONDITION MET
C
70    MLM=ML
      TRAD=THTAF*RAD
C
      VAR(5)=TRAD
      VAR(6)=RADIUS
      RETURN
      END
```

```
      SUBROUTINE JINIT
C
C INITIALIZES THE ARRAYS FOR THE METHOD OF CHARACTERISTICS
C TO SOLVE FOR ACTUAL PLUME SHAPE
C
      REAL M(21,11),L0,L9,M1,M6,MA(3,45)
      DIMENSION X(21,11),R(21,11),T(21,11)
     *,XH(11),XI(11),XJ(11),XK(11)
     *,RA(3,45),XA(3,45),TA(3,45)
      COMMON/C1005/XH,XI,XJ,XK
      COMMON/INJET/MEXIT,M6,K0,N0
      COMMON/ARRAY/X,R,M,T
      COMMON/SHAPE/X7,C0,R7,T0,R0,G0,XM0
      COMMON/INDICE/I,N,K,IS,IP
      COMMON/MARRAY/XA,MA,RA,TA
C
C  SET INITIAL CHARACTERISTIC VALUES FROM THE NORMALIZED
C  NOZZLE EXIT CONDITIONS.
C
C
      DO 100 J=1,K0
      X(1,J)=XA(1,J)
      R(1,J)=RA(1,J)
      M(1,J)=MA(1,J)
      T(1,J)=TA(1,J)
100   CONTINUE
C
      L0=SQRT((G0+1.)/(G0-1.))
      L9=(FNO(FNB(M6,G0),L0,G0)-FNO(FNB(XM0,G0),L0,G0)+T(1,1))*57.295
      WRITE(6,10)L9
10    FORMAT(///T50'PRANDTL-MEYER EXPANSION',
     *//T10'INITIAL SLOPE (DEG) = ',F10.4)
      I=1
      IP=2
      XH(1)=0.0
      XI(1)=1.
      XJ(1)=L9*3.14159/180.
      XK(1)=0.0
C
C  SETS UP INITIAL VALUES IN ARRAYS BASED ON EXPANSION AT LIP
C  FROM LIP EXIT MACH NUMBER TO JET SURFACE MACH NUMBER.
C  THE I VALUES ARE INCREMENTED FROM 1 TO NUMBER OF NET POINTS.
C  ALL INITIAL X VALUES = 0 AND ALL INITIAL R VALUES =1
C
      WRITE(6,40)
40    FORMAT(1H1//,T25'INITIAL VALUES FOR METHOD OF CHARACTERISTICS',
     */T25'TO SOLVE ACTUAL PLUME SHAPE')
      J=1
      STEP=(M6-XM0)/(FLOAT(N-1))-.0001
      DO 20 XM=XM0,M6,STEP
      M1=FNB(XM,G0)
      O=FNO(M1,L0,G0)
      OD=O*57.3
      WRITE(6,30)XM,M1,O,OD
30    FORMAT(7X,'M= 'F10.4,' M*= 'F10.4,' OMEGA ='F10.4,' DEG= 'F10.4)
      X(I,J)=0.
```

```
      R(I,J)=1.
      M(I,J)=M1
      T(I,J)=O-FNO(FNB(XMO,GO),LO,GO)+T(1,1)
      I=I+1
20    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE KLGLEV(X,Y,J)
C
C  THIS SUBROUTINE SOLVES THE TRANSONIC FLOW REGION IN THE
C  NOZZLE THROAT SECTION.  CALCULATIONS ARE MADE BEGINNING AT
C  THE THROAT (X=0,R=1)AND CONTINUING UNTIL THE MACH NUMBER
C  EXCEEDS 1.025 AT WHICH POINT THE PROGRAM SWITCHES TO
C  METHOD OF CHARACTERISTICS SOLUTION TO SOLVE FLOW FIELD OUT
C  TO THE NOZZLE EXIT.
C
      REAL*8 L1,L2,L3,L4,L5,N1,N2,N3,N4,N5,N6,Z0,Z7,Z8,Z9
      DIMENSION B(45),C(45)
      COMMON/CKLEV/B,C
      COMMON/SHAPE/X7,C0,R7,T0,R0,G0,XM0
      WRITE(6,7)
7     FORMAT(/T20'***  KLGLEV  ***')
      Z0=X*SQRT(2.*C0/(G0+1.))
      L1=Y*Y/2.-.25+Z0
      L2=((2.*G0+9.)*Y**4-(4.*G0+15.)*Y*Y)/24.
      L2=L2+(10.*G0+57.)/288.+Z0*(Y*Y-.625)-(2.*G0-3.)*Z0*Z0/6.
      L3=(556.*G0**2+1899.*G0+3231.)*Y**6/10368.-(388.*G0*G0
     &+1233.*G0+1953.)*Y**4
      L3=L3/2304.
      L3=L3+(304.*G0*G0+858.*G0+1269.)*Y*Y/1728.-(2708.*G0*G0+
     &7839.*G0+14211.)
      L3=L3/82944.
      L4=(52.*G0*G0+99.*G0+375.)*Y**4/384.-(52.*G0*G0+99.*G0+303.)
     &*Y*Y/192.+(92.*G0*G0+180.*G0+639.)/1152.
      L5=((13.*G0-27.)/48.-(5.*G0-5.)*Y*Y/8.)*Z0*Z0+Z0*Z0*Z0*
     *((4.*G0*G0-57.*G0+27.)/144.)
      L3=L3+Z0*L4+L5
      B(J)=1.+L1/(C0+1.)+(L1+L2)/(C0+1.)**2+(L1+2.*L2+L3)/
     &(C0+1.)**3
      N1=Y*Y*Y/4.-Y/4.+Y*Z0
      N2=(8.*G0+15.)*Y**5/72.-(20.*G0+45.)*Y**3/96.+
     &(28.*G0+75.)*Y/288.
      N2=N2+Z0*((4.*G0+9.)*Y**3/12.-(4.*G0+9.)*Y/12.)
      Z8=Y**5/13824.
      Z7=6836.*G0*G0
      N3=(Z7+16695.*G0+14211.)*Y**7/82944.-(3380.*G0*G0+
     &8703.*G0+7875.)*Z8
      N3=N3+(3424.*G0*G0+9183.*G0+8964.)*Y**3/13824.
      N3=N3-(7100.*G0*G0+19575.*G0+20745.)*Y/82944.
      N4=(556.*G0*G0+1113.*G0+981.)*Y**5/1728.-
     &(388.*G0*G0+801.*G0+693.)*Y**3/576.
      N4=N4+(304.*G0*G0+645.*G0+549.)*Y/864.
      N5=Z0*Z0*((52.*G0*G0+3.*G0-33.)*Y**3/192.-
     &(52.*G0*G0+27.*G0-9.)*Y/192.)
      N6=Z0**3*(G0+1.)*Y/4.
      N3=N3+Z0*N4+N5-N6
      Z9=SQRT((G0+1.)/(2.*(C0+1.)))
      Z8=1.5*N1+N2
      C(J)=Z9*(N1/(C0+1.)+(Z8)/(C0+1.)**2+(15.*N1/8.+
     &2.5*N2+N3)/(C0+1.)**3)
      RETURN
      END
```

```
      SUBROUTINE LISTPL(THTAF,RADIUS)
C
C
C THIS SUBROUTINE PRINTS THE PLUME SHAPE
C
C
      WRITE(8,15)
      WRITE(9,10)
10    FORMAT(1H1,//,40X,'PLUME SHAPE CALCULATED FROM JOHANNESEN'
     *'-MEYER THEORY'
     */T56,'X',T66,'R',T76,'THETA')
15    FORMAT(1H1,//,5X,'PLUME SHAPE CALCULATED FROM JOHANNESEN'
     *'-MEYER THEORY'
     */T21,'X',T31,'R',T41,'THETA')
      RAD2=0.0
      DO 30 S=0.,5.,.1
      X=SIN(THTAF)-(S/RADIUS)
      T=XMANG1(X)
      S1=RADIUS*(SIN(THTAF)-SIN(T))
      R=1.+RADIUS*(COS(T)-COS(THTAF))
      TDEG=T*57.3
      WRITE(8,25) S1,R,TDEG
      WRITE(9,20) S1,R,TDEG
20    FORMAT(T51,3F10.4)
25    FORMAT(T16,3F10.4)
      IF(RAD2.GT.R)RETURN
      RAD2=R
30    CONTINUE
      END
C
C****************************************************************
C
```

```
      SUBROUTINE LISTPR(MF,GAMMA)
      REAL LAMDA,MF
      COMMON/VAR/VAR(17)
C
C  THIS SUBROUTINE CALCULATES AND PRINTS THE PRESSURE RATIO
C
      LAMDA=SQRT((GAMMA-1.)/(GAMMA+1.))
      X=XMSQR(MF,GAMMA)
      P=(1.-X*LAMDA*LAMDA)**(GAMMA/(GAMMA-1.))
      VAR(7)=P
      RETURN
      END
```

```
      SUBROUTINE MODIN(GAMM,MFM,GAMP,MFP,IERR)
      REAL MFM
      DIMENSION HOL(20),FLO(20)
      COMMON/VAR/VAR(17)
C
C  THIS SUBROUTINE ALLOWS INPUT OF MODEL GAMMA
C  AND MODEL FINAL MACH NUMBER
C
      WRITE(1,10)
10    FORMAT(/T20'ENTER GAMMA OF MODEL')
      CALL FFREAD(HOL,FLO,LH,LF)
      GAMM=FLO(1)
C
      WRITE(1,20)
20    FORMAT(/T20'ENTER :  1-INPUT FINAL MACH OF MODEL'
     #/T29'2-CALCULATE WITH WEAK SHOCK RELATIONS'
     #/T29'3-CALCULATE WITH STRONG SHOCK RELATIONS')
      CALL FFREAD(HOL,FLO,LH,LF)
      IOPT=FLO(1)
      GO TO(30,40,50),IOPT
30    WRITE(1,22)
22    FORMAT(/T20'ENTER FINAL MACH NUMBER FOR MODEL')
      VAR(16)=MFM
      RETURN
C
40    CALL WEAKFM(GAMP,MFP,GAMM,MFM,IERR)
      VAR(16)=MFM
      RETURN
C
50    CALL STRGFM(GAMP,MFP,GAMM,MFM,IERR)
      VAR(16)=MFM
      RETURN
      END
```

```
      SUBROUTINE NETPT
C
C  THIS SUBROUTINE RETURNS THE INTERMEDIATE MOC NET
C  VALUES (X3,R3,XM3,T3) BASED ON THE 1 AND 2 VALUES
C
      COMMON/CBKPTS/X1,X2,R1,R2,XM1,XM2,T1,T2
      COMMON/CNETPT/X3,R3,XM3,T3
      COMMON/LASTV/XM5,T5,A5
      COMMON/SHAPE/X7,C0,R7,T0,R0,G0,XM0
C
      WRITE(6,7)
7     FORMAT(/T20'***  NETPT  ***')
      XM9=1.01
C
C  THE A VALUES ARE MACH ANGLES
C
      A1=XMANG2(XMLOC(XM1,G0))
      A2=XMANG2(XMLOC(XM2,G0))
      A4=A1
      A5=A2
      T4=T1
      T5=T2
      XM4=XM1
      XM5=XM2
      R4=R1
      R5=R2
C
C X3 AND R3 ARE THE POSITION OF THE INTERMEDIATE NET POINT
C
1530  X3=(R2-R1+X1*TAN(T4-A4)-X2*TAN(T5+A5))
     &/(TAN(T4-A4)-TAN(T5+A5))
      R3=R1+(X3-X1)*TAN(T4-A4)
C
      Q1=FNQ(A4,XM4)
      G1=FNG(A4,T4)
C
      Q2=FNQ(A5,XM5)
C
C  CHECK IF POINT IS ON CENTER LINE
C
      IF(R2.EQ.0.0)GO TO 2630
      F2=FNF(A5,T5)
C
C XM3 IS THE MACH NUMBER OF THE INTERMEDIATE NET POINT
C T3  IS THE THETA OF THE INTERMEDIATE NET POINT
C
      XM3=((T1-T2)+G1*(R3-R1)/R4+F2*(R3-R2)
     &/R5+Q1*XM1+Q2*XM2)/(Q1+Q2)
      T7=Q2*(XM3-XM2)-F2*(R3-R2)/R5
1680  T3=T2+T7
C
C  CHECK IF LOCAL MACH NUMBER CLOSE TO SONIC
C
      IF(ABS(XM9-XM3).LT..0001)RETURN
      XM9=XM3
      T4=(T1+T3)/2.
```

```
      T5=(T2+T3)/2.
      XM4=(XM1+XM3)/2.
      XM5=(XM2+XM3)/2.
      A3=XMANG2(XMLOC(XM3,G0))
      A4=(A1+A3)/2.
      A5=(A2+A3)/2.
      R5=(R3+R2)/2.
      R4=(R3+R1)/2.
      GO TO 1530
C
C  POINT IS ON CENTERLINE CALUCULATE MACH AND THETA
C
2630  XM3=(Q2*XM2/2.+T1+Q1*XM1+G1*(R3-R1)/R4)/
     &(Q2/2.+Q1)
      T7=Q2*(XM3-XM2)/2.
      GO TO 1680
      END
```

```
      SUBROUTINE NOZEXT(I,MEXIT)
      REAL M(3,45)
      DIMENSION X(3,45),R(3,45),T(3,45)
      COMMON/MARRAY/X,M,R,T
      COMMON/C1820/X8,R8,XM8,T8
      COMMON/SHAPE/X7,CO,R7,TO,RO,GO,XMO
      COMMON/VAR/VAR(17)
C
C  AFTER NOZZLE EXIT CONDITION PASSED,  MODIFY CHARACTERISTICS
C  TO LIP OF NOZZLE.
C
      W1=0.0
10    IF(MEXIT.EQ.0)WO=(RO-R(I-1,1))/(R(I+1,1)-R(I-1,1))
      IF(MEXIT.EQ.1)WO=(XMO-XMLOC(M(I-1,1),GO))/
     *(XMLOC(M(I+1,1),GO)-XMLOC(M(I-1,1),GO))
C
C  LINEARLY INTERPOLATE EXIT CONDITIONS
C
      X(I,1)=X(I-1,1)+(X(I,1)-X(I-1,1))*WO
      R(I,1)=R(I-1,1)+(R(I,1)-R(I-1,1))*WO
      M(I,1)=M(I-1,1)+(M(I,1)-M(I-1,1))*WO
      T(I,1)=T(I-1,1)+(T(I,1)-T(I-1,1))*WO
C
      CALL WALL(I)
      WOSW1=ABS(WO-W1)
C
      WRITE(6,769)WO,W1,WOSW1
769   FORMAT(5X,3F15.5)
      IF(ABS(WO-W1).LT. .01)GO TO 20
      X(I+1,1)=X8
      R(I+1,1)=R8
      M(I+1,1)=XM8
      T(I+1,1)=T8
      W1=WO
      GOTO 10
C
20    XM88=XMLOC(XM8,GO)
      T8D=T8*57.3
      VAR(13)=X8
      VAR(14)=R8
C
      RETURN
      END
```

```
      SUBROUTINE PLULIS
      COMMON/VAR/VAR(17)
C
C  THIS SUBROUTINE OUTPUTS THE PROGRAM INPUTS AND RESULTS
C
      WRITE(9,10)(VAR(I),I=1,7)
10    FORMAT(1H1,T56,'MODEL NOZZLE DESIGN'
     1//T45'PROTOTYPE'
     2/ T45'---------'
     3/T42'SPECIFIC HEAT RATIO.............................'F8.3,
     4/T42'NOZZLE ANGLE....................................'F8.3,
     5/T42'EXIT MACH NUMBER................................'F8.3,
     6/T42'JET SURFACE MACH NUMBER.........................'F8.3,
     7/T42'INITIAL SLOPE OF JET PLUME......................'F8.3,
     8/T42'INITIAL RADIUS OF CURVATURE OF JET PLUME........'F8.3,
     9/T42'PRESSURE RATIO..................................'F8.3)
      WRITE(9,20)(VAR(I),I=8,16)
20    FORMAT(//,T45,'MODEL',/,T45,'-----',
     1/T42'SPECIFIC HEAT RATIO.............................'F8.3,
     2/T42'THROAT RADIUS OF CURVATURE......................'F8.3,
     3/T42'BEGINNING AXIAL LOCATION OF CONICAL SECTION.....'F8.3,
     4/T42'BEGINNING RADIAL LOCATION OF CONICAL SECTION....'F8.3,
     5/T42'NOZZLE ANGLE, DEG...............................'F8.3,
     6/T42'NOZZLE LENGTH...................................'F8.3,
     7/T42'NOZZLE EXIT RADIUS..............................'F8.3,
     8/T42'EXIT MACH NUMBER................................'F8.3,
     9/T42'JET SURFACE MACH NUMBER.........................'F8.3)
      IF(VAR(17).LE. 2.)GO TO 50
      WRITE(8,30)(VAR(I),I=1,7)
30    FORMAT(1H1,T21,'MODEL NOZZLE DESIGN'
     1//T10'PROTOTYPE'
     2/ T10'---------'
     3/T7'SPECIFIC HEAT RATIO.............................'F8.3,
     4/T7'NOZZLE ANGLE....................................'F8.3,
     5/T7'EXIT MACH NUMBER................................'F8.3,
     6/T7'JET SURFACE MACH NUMBER.........................'F8.3,
     7/T7'INITIAL SLOPE OF JET PLUME......................'F8.3,
     8/T7'INITIAL RADIUS OF CURVATURE OF JET PLUME........'F8.3,
     9/T7'PRESSURE RATIO..................................'F8.3)
      WRITE(8,40)(VAR(I),I=8,16)
40    FORMAT(//,T10,'MODEL',/,T10,'-----',
     1/T7'SPECIFIC HEAT RATIO.............................'F8.3,
     2/T7'THROAT RADIUS OF CURVATURE......................'F8.3,
     3/T7'BEGINNING AXIAL LOCATION OF CONICAL SECTION.....'F8.3,
     4/T7'BEGINNING RADIAL LOCATION OF CONICAL SECTION....'F8.3,
     5/T7'NOZZLE ANGLE, DEG...............................'F8.3,
     6/T7'NOZZLE LENGTH...................................'F8.3,
     7/T7'NOZZLE EXIT RADIUS..............................'F8.3,
     8/T7'EXIT MACH NUMBER................................'F8.3,
     9/T7'JET SURFACE MACH NUMBER.........................'F8.3)
      VAR(5)=VAR(5)/57.3
50    CALL LISTPL(VAR(5),VAR(6))
      CALL WRITPL
      RETURN
      END
```

```
      SUBROUTINE PLUMIN(THTAFP,RADP)
      DIMENSION HOL(20),FLO(20)
      COMMON/RADS/PI,RAD
      COMMON/VAR/VAR(17)
C
C  THIS SUBROUTINE ALLOWS INPUT OF PLUME SHAPE
C
C  THTAFP=THETA-F OF PROTOTYPE
C  RADP=INITIAL RADIUS OF PROTOTYPE
C
      WRITE(1,10)
10    FORMAT(/T20'ENTER THETA-F AND INITIAL RADIUS'
     */T22'OF PROTOTYPE')
C
      CALL FFREAD(HOL,FLO,LH,LF)
      THTAFP=FLO(1)
      RADP=FLO(2)
      IF(LF.EQ.2) GO TO 20
      CALL FFREAD(HOL,FLO,LH,LF)
      RADP=FLO(1)
C
20    VAR(5)=THTAFP
      VAR(6)=RADP
C
      THTAFP=THTAFP/RAD
C
      RETURN
      END
```

```
      SUBROUTINE PLUMPT
      REAL M(21,11),M1,M2,M3,M5,M6
      DIMENSION X(21,11),R(21,11),T(21,11),XH(11),XK(11)
     *,XI(11),XJ(11)
      COMMON/INJET/MEXIT,M6,KO,NO
      COMMON/ARRAY/X,R,M,T
      COMMON/C1005/XH,XI,XJ,XK
      COMMON/INDICE/I,N,K,IS,IP
      COMMON/SHAPE/X7,CO,R7,TO,RO,GO,XMO
      J=1
      X2=X(N,J+1)
      R2=R(N,J+1)
      M2=M(N,J+1)
      T2=T(N,J+1)
      A2=XMANG2(XMLOC(M2,GO))
      A3=XMANG2(XMLOC(M(N,J),GO))
      R1=R(N,J)
      X1=X(N,J)
      M1=M(N,J)
      T1=T(N,J)
      A5=A2
      M3=FNB(M6,GO)
      A3=XMANG2(XMLOC(M3,GO))
      T5=T2
      M5=M2
      R5=R2
      T9=T1
      X3=X2
      X3=(R1-R2+X2*TAN(T5+A5)-X1*TAN(T9))/(TAN(T5+A5)-TAN(T9))
1105  R3=R2+(X3-X2)*TAN(T5+A5)
      Q2=1./(M5*TAN(A5))
      F2=SIN(A5)*SIN(T5)/SIN(A5+T5)
      T3=T2+Q2*(M1-M2)-F2*(R3-R2)/R5
      T9=(T1+T3)/2.
      X7=(R1-R2+X2*TAN(T5+A5)-X1*TAN(T9))/(TAN(T5+A5)-TAN(T9))
      IF(ABS(X3-X7) .LT. .001)GO TO 1170
      X3=X7
      T5=(T2+T3)/2.
      M5=(M2+M3)/2.
      A5=(A2+A3)/2.
      R5=(R2+R3)/2.
      GOTO 1105
1170  RO=(R1-R3)/(COS(T1)-COS(T3))
C
      T3D=T3*57.3
      X(N+1,1)=X3
      XH(IP)=X3
      XK(IP)=RO
      R(N+1,1)=R3
      XI(IP)=R3
      M(N+1,1)=M3
      T(N+1,1)=T3
      XJ(IP)=T3
      RETURN
      END
```

```
      SUBROUTINE PRTOPT
      DIMENSION HOL(20),FLO(20)
      COMMON/VAR/VAR(17)
C
C  SETS UP FILES AND LOGICAL UNITS FOR OUTPUT
C
1     WRITE(1,10)
10    FORMAT(/T20'ENTER OUTPUT PREFERENCE'/,
     */T20'PRINTER..................1',
     */T20'VARIAN...................2',
     */T20'CONSOLE..................3',
     */T20'CONSOLE AND PRINTER......4',
     */T20'CONSOLE AND VARIAN.......5')
C
      CALL FFREAD(HOL,FLO,LH,LF)
      IPRTOP=FLO(1)
      VAR(17)=IPRTOP
      IF(IPRTOP.NE.1 .AND. IPRTOP.NE.4)GO TO 20
      CALL CLOSE(9,ISTAT)
      CALL OPENW(9,'PR:',2,0,0,ISTAT)
      IF(IPRTOP.EQ.1)GO TO 100
C
20    IF(IPRTOP.NE.2 .AND. IPRTOP.NE.5)GO TO 30
      CALL CLOSE(9,ISTAT)
      CALL OPENW(9,'PLOT:',2,0,0,ISTAT)
      IF(ISTAT.NE.0)GO TO 40
      IF(IPRTOP.EQ.5)GO TO 30
      GO TO 100
40    WRITE(1,22)
22    FORMAT(/T15'** ERROR WITH VARIAN - CHECK AND RE-ENTER**')
      GO TO 1
C
30    CALL CLOSE(8,ISTAT)
      CALL OPENW(8,'CON:',4,0,0,ISTAT)
C
100   RETURN
      END
```

```
      SUBROUTINE PRWALL(I)
C
C  THIS SUBROUTINE PRINTS OUT CONDITIONS AT POINTS
C  ALONG THE WALL(PRINTS TO UNIT 6)
C
      REAL M(3,45)
      DIMENSION R(3,45),T(3,45),X(3,45)
      COMMON/MARRAY/X,M,R,T
      COMMON/SHAPE/X7,CO,R7,TO,RO,GO,XMO
C
      WRITE(6,7)
7     FORMAT(/T20'***  PRWALL  ***')
      D1=XMANG2(XMLOC(M(I-1,1),GO))
      D2=XMANG2(XMLOC(M(I,1),GO))
      D3=-COS(D1)+COS(D2)*M(I,1)/M(I-1,1)
      D4=SQRT((X(I-1,1)-X(I,1))**2+
     *(R(I-1,1)-R(I,1))**2)/R(I-1,1)
C
      U1=D3*M(I-1,1)/D4
      D5=D3/D4+SIN(TO)*(SIN(D2))**2
      D6=D5/(2.*(COS(D2))**2)
C
      ROR=R(I-1,1)
      XOR=X(I-1,1)
C
      WRITE(6,10)U1,D6,ROR,XOR
10    FORMAT(/T10,'U1=',F10.4,5X,'ACCEL=',F10.4,
     */T10'AT WALL POINT WHERE R/R* =',F10.4,
     */T10'AND                 X/R  =',F10.4)
C
      RETURN
      END
```

```
      SUBROUTINE RESET(K,S)
C
C  THIS SUBROUTINE STORES THE VALUES OF THE CHARACTERISTIC
C  POINTS JUST CALCULATED AND CURRENTLY IN THE I=3 POSITION
C  INTO THE I=1 POSITION.  THIS SUBROUTINE CALLED EVERY OTHER
C  PASS.  THEREFORE, TWO PLANES OF CHARACTERISTICS ARE CARRIED
C  ALONG ALWAYS AND THE THIRD PLANE ONLY TEMPORARILY.
C
      REAL M(3,45)
      INTEGER S
      DIMENSION X(3,45),R(3,45),T(3,45)
      COMMON/MARRAY/X,M,R,T
      WRITE(6,7)
7     FORMAT(/T20'***  RESET  ***')
C
      DO 10 J=1,K-S+1
      X(1,J)=X(3,J)
      R(1,J)=R(3,J)
      M(1,J)=M(3,J)
      T(1,J)=T(3,J)
10    CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE SETEXT(P,A8,I,Q,K,S,N,IFINI)
C
C
      REAL M(3,45)
      INTEGER S,P,A8,Q
      DIMENSION D(3,45),E(3,45),F(3,45),G(3,45)
     *,X(3,45),R(3,45),T(3,45)
      COMMON/MARRAY/X,M,R,T
      COMMON/C2480/D,E,F,G
C
C  SETS UP CHARACTERISTICS AFTER EXIT REACHED
C
C  NOTE: P<0 AFTER EXIT.  Q=0 UNTIL EXIT THEN INCREASED
C  BY 1 EVERY OTHER PASS
      WRITE(6,7)
7     FORMAT(/T20'***  SETEXT  ***')
C
      IF(A8.EQ.2. .AND.P.LT.0.) Q=Q+1
      I1=I+1
      WRITE(6,10)I1
10    FORMAT(/T20'ROW (I+1)=',I4)
      DO 20 J=1,K-S
      IF(J.LT.Q)GO TO 20
      IF(J.NE.Q.OR.P.GE.0)GO TO 2540
      N=N+1
      D(1,N)=X(I+1,J)
      E(1,N)=R(I+1,J)
      F(1,N)=M(I+1,J)
      G(1,N)=T(I+1,J)
C
C
2540  TDEG=T(I1,J)*57.3
      WRITE(6,30)J,X(I1,J),R(I1,J),M(I1,J),TDEG
30    FORMAT(1X,I4,4F10.3,2X,'SETEXT')
      IF((2*(K-S)-1).EQ.N)GO TO 40
20    CONTINUE
      RETURN
40    CALL STOREX(K,S)
      IFINI=1
      RETURN
      END
C
C
C
C
```

```
      SUBROUTINE SETPTS(K,S,I)
C
C THIS SUBROUTINE SETS UP THE TWO POINTS NEEDED TO
C  CALCULATE THE NEXT INTERMEDIATE NET POINT
C
      INTEGER S
      REAL M(3,45)
      DIMENSION X(3,45),T(3,45),R(3,45)
C
      COMMON/MARRAY/X,M,R,T
      COMMON/CBKPTS/X1,X2,R1,R2,XM1,XM2,T1,T2
      COMMON/CNETPT/X3,R3,XM3,T3
C
      WRITE(6,7)
7     FORMAT(/T20'***  SETPTS  ***')
C
C
      DO 10 J=1,K-S
      X1=X(I,J)
      X2=X(I,J+1)
      R1=R(I,J)
      R2=R(I,J+1)
      XM1=M(I,J)
      XM2=M(I,J+1)
      T1=T(I,J)
      T2=T(I,J+1)
      CALL NETPT
      X(I+1,J)=X3
      R(I+1,J)=R3
      M(I+1,J)=XM3
      T(I+1,J)=T3
10    CONTINUE
C
C
C
      RETURN
      END
```

```
      SUBROUTINE SETPT2
      REAL M(21,11),M1,M2,M3
      DIMENSION X(21,11),R(21,11),T(21,11)
      COMMON/CBKPTS/X1,X2,R1,R2,M1,M2,T1,T2
      COMMON/CNETPT/X3,R3,M3,T3
      COMMON/ARRAY/X,R,M,T
      COMMON/INDICE/I,N,K,IS,IP
      DO 10 J=1,K-1
       X1=X(I+1,J)
       X2=X(I,J+1)
       R1=R(I+1,J)
       R2=R(I,J+1)
       M1=M(I+1,J)
       M2=M(I,J+1)
       T1=T(I+1,J)
       T2=T(I,J+1)
      CALL NETPT
       X(I+1,J+1)=X3
       R(I+1,J+1)=R3
       M(I+1,J+1)=M3
       T(I+1,J+1)=T3
10    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SETPT3
      REAL M(21,11),M1,M2,M3
      DIMENSION X(21,11),R(21,11),T(21,11)
      COMMON/CNETPT/X3,R3,M3,T3
      COMMON/CBKPTS/X1,X2,R1,R2,M1,M2,T1,T2
      COMMON/ARRAY/X,R,M,T
      COMMON/INDICE/I,N,K,IS,IP
      DO 10 J=1,K-2
      X1=X(N+1,J)
      X2=X(N,J+2)
      R1=R(N+1,J)
      R2=R(N,J+2)
      M1=M(N+1,J)
      M2=M(N,J+2)
      T1=T(N+1,J)
      T2=T(N,J+2)
      IF(M2.LT. 1.)RETURN
      CALL NETPT
      X(N+1,J+1)=X3
      R(N+1,J+1)=R3
      M(N+1,J+1)=M3
      T(N+1,J+1)=T3
10    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE STOREX(K,S)
C
C  CALLED AFTER SUFFICIENT CHARACTERISTICS PAST LIP
C  ARE CALCULATED.  THE EXIT CHARACTERISTICS ARE
C  NORMALIZED AND STORED.
C
      INTEGER S
      REAL M(3,45)
      DIMENSION X(3,45),T(3,45),R(3,45),D(3,45),E(3,45),F(3,45)
     *,G(3,45)
C
      COMMON/C2480/D,E,F,G
      COMMON/MARRAY/X,M,R,T
      COMMON/SHAPE/X7,CO,R7,TO,RO,GO,XMO
C
      WRITE(6,7)
7     FORMAT(/T20'***  STOREX  ***')
      WRITE(6,10)
10    FORMAT(/T20'NOZZLE EXIT CHARACTERISTIC NORMALIZED AND STORED')
      WRITE(6,20)
20    FORMAT(/T23'N'T30'X(1,N)'T40'R(1,N)'T50'M(1,N)'T60'T(1,N) RAD')
C
      KS2=2*(K-S)-1
      DO 30 N=1,KS2
      X(1,N)=(D(1,N)-D(1,1))/E(1,1)
      R(1,N)=E(1,N)/E(1,1)
      T(1,N)=G(1,N)
      M(1,N)=F(1,N)
C
      WRITE(6,40)N,X(1,N),R(1,N),M(1,N),T(1,N)
C
30    CONTINUE
C
      EXITM=XMLOC(M(1,1),GO)
      WRITE(6,60)EXITM
60    FORMAT(/10X,'EXIT MACH NUMBER AT LIP = ',F12.4)
40    FORMAT(I24,4F10.3)
C
      RETURN
      END
```

```
      SUBROUTINE STRGFM(GAMP,MFP,GAMM,MFM,IERR)
      REAL MFP,MFM
C
C  SOLVES FOR THE FINAL MACH NUMBER OF THE MODEL, ALSO
C  REFERRED TO AS THE JET SURFACE MACH NUMBER.
C  THE STRONG SHOCK APPROXIMATION FOR MATCHING THE SUPERSONIC
C  INVISCID STREAMLINE DEFLECTION-PRESSURE RISE RELATION IS USED.
C
      A=(2.*GAMP*MFP*MFP-GAMP+1.)/(GAMP+1.)
      B=A*(GAMM+1.)+GAMM-1.
      C=B/(2.*GAMM)
      IF(C.GT.0.0)GO TO 10
      WRITE(1,20)GAMP,GAMM,MFP
20    FORMAT(//T10'** A  SOLUTION FOR THE JET SURFACE MACH NUMBER'
     */T10'DOES NOT EXIST FOR THE SPECIFIED CONDITIONS**'
     */T15'GAMMA PROTOTYPE = 'F10.2,
     */T15'GAMMA MODEL     = 'F10.2,
     */T15'MACH  MODEL     = 'F10.2)
      IERR=1
      RETURN
C
10    MFM=SQRT(C)
C
      RETURN
      END
```

```
      SUBROUTINE WALL(I)
C
C  THIS SUBROUTINE SOLVES FOR THE WALL BOUNDARY
C
      REAL M(3,45)
      DIMENSION X(3,45),R(3,45),T(3,45)
      COMMON/MARRAY/X,M,R,T
      COMMON/SHAPE/X7,CO,R7,TO,RO,GO,XMO
      COMMON/LASTV/XML,TL,AL
      COMMON/C1820/X8,R8,XM8,T8
C
      WRITE(6,7)
7     FORMAT(/T20'***  WALL  ***')
C
      X2=X(I,1)
      R2=R(I,1)
      XM2=M(I,1)
      T2=T(I,1)
      A2=XMANG2(XMLOC(XM2,GO))
      XM9=1.01
      A5=A2
      T5=T2
      XM5=XM2
      R5=R2
1920  T9=TAN(T5+A5)
C
C  CHECK IS CONICAL SECTION REACHED
C
      IF(X2.LE.X7.AND.X3.LE.X7)GO TO 2060
C
C  CALCULATE X,R,THETA IN CONICAL SECTION
C
2590  X3=(R2-R7+X7*TAN(TO)-X2*T9)/(TAN(TO)-T9)
      R3=R7+(X3-X7)*TAN(TO)
      T3=TO
      GO TO 2121
C
C  CALCULATE X,R,THETA IN CIRCULAR THROAT SECTION
C
2060  XO=CO+1.-R2+T9*X2
      X9=T9*XO/(1.+T9*T9)
      X3=X9-SQRT(X9*X9+(CO*CO-XO*XO)/(T9*T9+1.))
      IF(X3.GT.X7)GO TO 2590
      R3=1.+CO*(1.-SQRT(1.-X3*X3/(CO*CO)))
      T3=ATAN(1./SQRT(CO*CO/(X3*X3)-1.))
      T3D=T3*57.2958
      WRITE(6,10)X3,R3,T3D
10    FORMAT(3F12.5,2X,'WALL')
2121  Q2=FNQ(A5,XML)
      F2=FNF(A5,TL)
      XM3=XM2+(T3-T2+F2*(R3-R2)/R5)/Q2
      IF(ABS(XM9-XM3).LT. .0001)GO TO 2220
      XM9=XM3
      T5=(T2+T3)/2.
      XM5=(XM2+XM3)/2.
      A3=XMANG2(XMLOC(XM3,GO))
```

```
      A5=(A2+A3)/2.
      R5=(R3+R2)/2.
      GO TO 1920
C
2220  X8=X3
      R8=R3
      XM8=XM3
      T8=T3
      RETURN
      END
```

```
      SUBROUTINE WEAKFM(GAMP,MFP,GAMM,MFM,IERR)
      REAL MFP,MFM
C
C  SOLVES FOR THE FINAL MACH NUMBER OF THE MODEL, ALSO
C  REFERRED TO AS THE JET SURFACE MACH NUMBER.
C  THE WEAK SHOCK APPROXIMATION FOR MATCHING THE SUPERSONIC
C  INVISCID STREAMLINE DEFLECTION-PRESSURE RISE RELATION IS USED.
C
      A=GAMP/GAMM*MFP*MFP/SQRT(MFP*MFP-1.)
      IF(A.GE.2.)GO TO 10
      WRITE(1,20)GAMP,GAMM,MFP
20    FORMAT(//T10'** A SOLUTION FOR THE JET SURFACE MACH NUMBER'
     */T10'DOES NOT EXIST FOR THE SPECIFIED CONDITIONS**'
     */T15'GAMMA PROTOTYPE = 'F10.2,
     */T15'GAMMA MODEL     = 'F10.2,
     */T15'MACH  MODEL     = 'F10.2)
      IERR=1
      RETURN
C
10    MFM=SQRT((A*A+A*SQRT(A*A-4.))/2.)
C
      RETURN
      END
```

```
      SUBROUTINE WRITPL
      DIMENSION XK(11),XI(11),XJ(11),XH(11)
      COMMON/INDICE/I,N,K,IS,IP
      COMMON/C1005/XH,XI,XJ,XK
C
C  WRITE OUT RESULTS
C
      WRITE(8,15)
      WRITE(9,10)
10    FORMAT(1H1,//T38'MODEL PLUME SHAPE CALCULATED BY METHOD OF'
     *' CHARACTERISTICS'
     */T56,'X',T66,'R',T76,'THETA')
15    FORMAT(1H1,//T2'MODEL PLUME SHAPE CALCULATED BY METHOD OF'
     *' CHARACTERISTICS'
     */T21,'X',T31,'R',T41,'THETA')
      DO 30 I=1,IP-1
      WRITE(8,25)XH(I),XI(I),XJ(I)*57.3
      WRITE(9,20)XK(I),XH(I),XI(I),XJ(I)*57.3
20    FORMAT(T51,3F10.4)
25    FORMAT(T16,3F10.4)
30    CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE WRITRW
C
C  CALLED TO WRITE OUT INTERMEDIATE VALUES  WRITES GO TO
C  LU 6  ONLY CALLED FOR DEBUGGING
C
      REAL M(21,11)
      DIMENSION X(21,11),R(21,11),T(21,11)
      COMMON/ARRAY/X,R,M,T
      COMMON/INDICE/I,N,K,IS,IP
C
      I1=I+1
      WRITE(6,10)I1
10    FORMAT(/T20'ROW = ',I3)
C
      DO 20 J=1,K
      TD=T(I1,J)*57.3
      WRITE(6,30)J,X(I1,J),R(I1,J),M(I1,J),TD
30    FORMAT(1X,I3,4F12.4)
20    CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE YINCR(P,MEXIT,I,K,S)
      REAL M(3,45)
      INTEGER S,P
      DIMENSION X(3,45),R(3,45),T(3,45)
      COMMON/MARRAY/X,M,R,T
      COMMON/C1820/X8,R8,XM8,T8
      COMMON/SHAPE/X7,CO,R7,TO,RO,GO,XMO
C
C  MOVES THE CURRENTLY CALCULATED CHARACTERISTIC POINTS UP ONE
C  IN THE Y POSITION OF THE ARRAYS (J)
C
      WRITE(6,7)
7     FORMAT(/T20'***  YINCR  ***')
      KS1=K-S+1
      DO 10 J=KS1,1,-1
      X(I+1,J+1)=X(I+1,J)
      R(I+1,J+1)=R(I+1,J)
      M(I+1,J+1)=M(I+1,J)
      T(I+1,J+1)=T(I+1,J)
10    CONTINUE
C
C  SET WALL CONDITIONS INTO J=1 POSITION
C
      X(I+1,1)=X8
      R(I+1,1)=R8
      M(I+1,1)=XM8
      T(I+1,1)=T8
C
C  CHECK FOR CONICAL SECTION
C
      K=K+2
      IF(X(I+1,1).LT.X7)GO TO 2410
      CALL PRWALL(I)
C
C  P=1 BEFORE NOZZLE EXIT AND P=-1 AFTER NOZZLE EXIT
C
2410  IF(P.LT.0)RETURN
C
C  TEST IF SPECIFIED EXIT MACH NUMBER EXCEEDED
C
      IF(MEXIT.NE.1)GOTO 40
      IF(XMLOC(M(I+1,1),GO).GT.XMO)GO TO 30
      RETURN
C
C  TEST IF SPECIFIED EXIT RADIUS EXCEEDED
C
40    IF(R(I+1,1).LE.RO)RETURN
C
30    P=-1
      CALL NOZEXT(I,MEXIT)
      RETURN
      END
```

```
      FUNCTION XMANG1(XMACH)
C
C  THIS FUNCTION RETURNS THE MACH ANGLE
C   (SAME AS SIN ALPHA=1/M)
C  THE INPUT XMACH IS 1/M
C
      XMANG1=ATAN(XMACH/SQRT(1.-XMACH*XMACH+1.E-50))
      RETURN
      END
C
C***********************************************************
C
      FUNCTION XMSQR(XMACH,GAMMA)
C
C  THIS FUNCTION CALCULATES THE VALUE OF M* SQUARED
C  WHERE M*=V/C*=V/V*.
C  NOTE: THIS ADIABATIC RELATIONSHIP CAN BE FOUND
C  IN CHAP. 4 PAGE 81 OF SHAPIRO'S COMPRESSIBLE FLUID FLOW
C
      XMSQR=(((GAMMA+1.)/2.)*XMACH*XMACH)/(1.+((GAMMA-1.)*XMACH*XMACH)
     &/2.)
      RETURN
      END
C
C***********************************************************
C
      SUBROUTINE TANGLE(XMACH,THETA,LAMDA)
      REAL LAMDA
C
C  THIS SUBROUTINE CALCULATES THE PRANDTL-MAYER FUNCTION
C
C  CHAP. 15 SHAPIRO
C
      XK=((1.-XMACH)/(XMACH-1./LAMDA/LAMDA))**(.5)
      THETA=ATAN(XK)/LAMDA-ATAN(XK/LAMDA)
      RETURN
      END
```

```
C
C*************************************************************
C
      FUNCTION XMANG2(X)
C
C THIS FUNCTION RETURNS THE MACH ANGLE
C  THE INPUT IS MACH
C
      XMANG2=ATAN(1./SQRT(X*X-1.))
      RETURN
      END
C
C*************************************************************
C
C
C THIS FUNCTION RETURNS LOCAL MACH NUMBER
C  THE INPUT IS M*
C
      FUNCTION XMLOC(X,G0)
      XMLOC=SQRT(2.)*X/SQRT(G0+1.-(G0-1.)*X*X)
      RETURN
      END
C
C*************************************************************
C
      FUNCTION FNQ(A,XM)
      FNQ=1./XM/TAN(A)
      RETURN
      END
```

```
C
C*************************************************************
C
      FUNCTION FNF(A,T)
      FNF=SIN(T)*SIN(A)/SIN(T+A)
      RETURN
      END
C
C*************************************************************
C
      FUNCTION FNG(A,T)
      FNG=SIN(T)*SIN(A)/SIN(T-A)
      RETURN
      END
C
C*************************************************************
C
      FUNCTION FNB(X,G0)
      FNB=X*SQRT((G0+1.)/(2.+(G0-1.)*X*X))
      RETURN
      END
```

```
C
C*********************************************** ***************
C
      FUNCTION FNC(X,GO)
      FNC=SQRT((X*X-1.)/((GO+1.)/(GO-1.)-X*X))
      RETURN
      END
C
C***************************************************************
C
      FUNCTION FNO(X,XLO,GO)
      FNO=XLO*ATAN(FNC(X,GO))-ATAN(XLO*FNC(X,GO))
      RETURN
      END
```

REFERENCES

1. Korst, H.H., "Approximate Determination of Jet Contours Near the Exit of Axially Symmetrical Nozzles as a Basis for Plume Modeling", U.S. Army Missile Command, Redstone Arsenal, Alabama Report No. TR-RD-72-14, August 1972.

2. Jonannesen, N.H. and Meyer, R.E., "Axially-Symmetrical Supersonic Flow Near the Centre of an Expansion," The Aeronautical Quarterly ,Vol. 2 ,1950, pp. 127-142.

3. Korst, H.H. and Deep, R.A., "Modeling of Plume Induced Interference Problems in Missile Aerodynamics," AIAA Paper No. 79036, 17th Aerospace Sciences Meeting, New Orleans, La., Jan. 1979.

4. Hall, I.M., "Transonic Flow in Two-Dimensional and Axially-Symmetric Nozzles," Quarterly Journal of Mechanics and Applied Mathematics , Vol. XV, Pt. 4, 1962,pp.487-508.

5. Kliegel, J.R. and Levine, J.N.,"Transonic Flow in Small Throat Radius of Curvature Nozzles," AIAA Journal ,Vol. 7, July 1969, pp. 1375-1378.

6. Dutton, J.C. and Addy, A.L., "Transonic Flow in the Throat Region of Axisymmetric Nozzles," AIAA Journal , Vol 19, No.6, June 1981,pp. 801-804.

7. Nyberg, S. -E., Agrell, J. and Hevreng, T., "Investigation of Modeling Concepts for Plume-Afterbody Flow Interactions," 2nd Annual Report, Aeronautical Research Institute of Sweden, European Research Office Grant No. DA-ERO-78-G-028, Feb. 1980.